

Ranking classifieds at Marktplaats.nl: Query Modeling, Retrieval Methods, Data Fusion, and Result Diversification

Varvara Tzika

June 24, 2014



Abstract



thesis

Our goal in this paper is to find an optimal solution to result a list with relevant classifieds. We derive multiple query models from a given classified, which are then used to retrieve similar classifieds from a classifieds index, resulting in multiple ranked lists. We then diversify this lists. Also we merge this initial lists as well as the diversified lists using data fusion techniques. Query models are created by exploiting the structure of the given classified and by discriminative terms of the classifieds context. For our experimental evaluation, we use data from an entrepreneurial database with users created classifieds for both query models and index. We show that using all the information we have available to the query modeling is producing a result list with high precision. Also, we proved that fusion techniques are not improving the performance of individual models. However, the fusion of a diversified results has the highest precision. Finally, we propose three alternative diversification methods.

Contents

1	Introduction	4
2	Related Work	7
2.1	Query modeling	7
2.2	Diversification	7
2.3	Data fusion	8
2.4	Data fusion of diversified result lists	8
2.5	Our contribution	9
3	Method	10
3.1	Query modeling	10
3.1.1	Source classified fields' query models	10
3.1.2	Pseudo Relevance Feedback	11
3.1.3	Log Likelihood Ratio	11
3.2	Retrieval modeling	13
3.2.1	TfIdf	13
3.2.2	Okapi BM25	13
3.2.3	Probabilistic Language Modeling	14
3.3	Diversification	15
3.3.1	Maximal marginal relevance alternative 1	15
3.3.2	Maximal marginal relevance alternative 2	16
3.3.3	Maximal marginal relevance average last four	16
3.4	Late Data Fusion	17
3.5	Data fusion of diversified result lists	17
4	Experiment Design	18
4.1	Data and data gathering	18
4.2	Results Retrieval	20
4.3	Our Baseline	22
4.4	Experiments	22
4.4.1	Stemming experiment-Experiment	22
4.4.2	Query modeling experiments	23
4.4.3	Retrieval methods	24
4.4.4	Late Data Fusion experiments	25
4.4.5	Diversification experiments	25
4.4.6	Data fusion of diversified result experiment	26
4.5	Evaluation	26
4.5.1	Assessors Evaluation	26

4.5.2	Clicks Evaluation	28
4.5.3	Measures	28
5	Results	29
5.1	Query modeling	29
5.2	Retrieval method	30
5.3	LDF	30
5.4	Diversification	31
5.5	Fused diversified results	32
6	Analysis	33
6.1	Stemming experiment	33
6.2	Query modeling	34
6.3	Retrieval methods	38
6.4	Late Data Fusion experiments	38
6.5	Diversification experiments	41
6.6	Data fusion of diversified result experiment	41
7	Conclusion and Future work	42
8	Appendix	48
8.1	Query modeling	48
8.1.1	Results Relevance ground truth	48
8.1.2	Results Click ground truth	49
8.2	LDF	49
8.2.1	LDF - Relevance Feedback	49
8.2.2	LDF - Clicks Ground Truth	50
8.2.3	combANZ - Click ground truth	50
8.2.4	combANZ - Relevance ground truth	50
8.3	LDF-MMR	51
8.3.1	LDF Results-MMR	51
8.3.2	LDF Results-MMRAlt1	51
8.3.3	LDF Results-MMRAlt2	51
8.3.4	LDF Results-MMRAltAvgLast4	52
8.4	Diversification	52
8.4.1	MMR	52
8.4.2	MMRAlt1	54
8.4.3	MMRAlt2	56
8.4.4	MMRAltAvgLst4	58

Chapter 1

Introduction

Nowadays, it is a demand of the market to provide to your users functionalities like “Recommended to you” or “Relevant advertisements” in case a company wants to keep users satisfied or increase sales. Furthermore, providing to the user a list with similar ads based on their interest can **improve the time** the user will spend to cover his information need. In our case, we are doing experiments based on an e-commerce platform’s data on which most content is in the form of advertisements created by common users selling a broad range of items called classifieds. A classified **is consisted** of content describing elements(fields) such as title, description and attributes that are created by common users.

~~Many companies had to come up with related solutions to problems like these, as Google News~~ **who** groups news by story rather than presenting a raw listing of all articles. Also, they record every click or search that every user made and just below the “Top Stories” section users can see a section labeled “Recommended for your email address” along with three stories that are recommended to them based on their past click history [48]. Likewise, Amazon.com based on users past shopping history and site activity recommends books and other items likely to be of interest. In the music industry, **[49]** use data from Last.fm and create user profiles based on the genres of the most listened artists. They create communities based on the categorization. **So,** for example **one** community is hip hop and the members all listen to hip hop. Also, Apples Genius feature in iTunes classifies songs into potential play lists for users [50]. The previous cases take advantage of a good structure of their big amount of data logs due to the categorization that they create. They use the history of users to predict users **interests** and give better recommendations. They increased their revenues, they keep their users satisfied and they eliminate the time that users spend to find the product that they want.

The big amount of information a classified has is what we need to relate users behavior with better recommendations. However, to show to user relevant classifieds based on their interest requires **finding** the relevant classifieds based on their history and **to relate** the information with other classifieds. One of the options is to classify them in categories **however,** trawling through hundreds or thousands of categories and subcategories of data is no longer an efficient method for finding information [51]. Other option is to use a recommendation algorithm based on other user’s history but then we have to deal also with new users that **have not any** history to relate with others. We could also use the click logs to improve the ranking of the search results. However, these methods rely on already optimal search algorithms that we are currently lacking. Therefore, our goal is first to introduce a set of optimal solutions for retrieving classifieds and then in future work we will explore click models and learning to rank approaches.

Recommending similar classifieds to users involves retrieving the most important information of the classified which already **had seen**. Classifieds have this information either on their

description or in their attributes. Extracting this knowledge from enormous amounts of data can be achieved with methods from the information retrieval area which is defined as “the area of study concerned with searching for documents, for information within documents, and for metadata about documents.”[52].

Our approach, described briefly, is to find an optimal algorithm which will create a list with similar classifieds based on the user’s last visited classified that represent his information need. As it is mentioned above, classifieds consists of content describing elements (fields) such as title, description, attributes and category. The most of the knowledge is in description. While title is a synoptic descriptive summary of all the information. Category is a broad category name provided by the user and attributes are based on the choice of the category. Examples of bicycles classified’s attributes are the color, the height, used condition etc. So, from the moment that a user will provide the category of the classified as bicycle, he has to specify the color, height and size. For this reason we are always considering the category and attributes as one field.

Such a problem belongs to the Information Retrieval domain. With the use of information retrieval we can extract terms of a classified that can represent the information need of the user. This will be our query and this procedure is the query modeling. Then with the use of a retrieval strategy, we will have as result the list with similar classifieds. Retrieval strategy is responsible to search in our indexed classifieds based on the input(query) and they will produce the list with similar classifieds(results list).

Query modeling has been a topic of active research for many years. One way to make better query models is to take feedback from users information need. In the absence of explicit user feedback, the canonical approach is to treat the last visited document retrieved in response to a query as it is users information need. We refer to this document as the visited classified.

A good choice of the retrieval models or the query modeling will be proved once adequate lists are constructed based on the input that we gave. A Different query model will result a different result list. A Different retrieval strategy will result different results list as well. The choice of the best query model and retrieval strategy are subjects for experimentation. The different results that the model will give us will affect the performance and the accuracy of the system. A good performing system or retrieval strategy will be evaluated based on the precision of the system. That means that it will be affected by the number of relevant results are retrieved by a system and the ordering(rank). System is the combination of the query modeling, retrieval strategy and all the systemic properties like how did we process the classified (e.g. removing noise words like ‘the’, ‘or’ etc.). Our work takes place in an enterprise setting, where data and different kind of users can help us to evaluate different query models.

Different query models will be created based on different combination of the fields. Also, we can extract discriminative terms of the visited classified to present the information need. Furthermore, we can take feedback from a result list and create a new query that it will result a new list.

Retrieval strategies exist since there is are existed from the early 90’s and no need to create a new one. However, we will investigate which is the best that it will increase our performance of our systems.

In a big amount of classifieds is possible that there are a lot of possible relevant, redundant or containing partially or fully duplicative information. Our goal is to expose less classifieds with high potential to cover the information need of the user. Thus, experiments will be conducted to find the best way to provide a diversified result list instead of a list with a lot of duplicated classifieds. Also, since user’s information need are often ambiguous, we can give to the user more diversified results to increase the possibility one of them to satisfy them.

To improve our results even further, we use late data fusion techniques. Merging the result

lists that are retrieved by multiple query models to one new list is proved that is improve the performance of the individual systems. Thus experiments with multiple late fusion techniques are provided. We also use fusion methods on the diversified results list to compare and see if any improvement occurs.

To conclude, we derive multiple query models from a given classified, which are then used to retrieve similar classifieds from a classifieds index, resulting in multiple ranked lists. We then diversify this lists. Also we merge this initial lists as well as the diversified lists using data fusion techniques. Query models are created by exploiting the structure of the given classified and by discriminative terms of the classifieds context. After all this mentioned we present the research questions we want to answer:



The research questions we aim to answer are the following:

1. Which query model is the best to improve the performance of title query model?
2. Which of the three retrieval strategies is performing better?
3. Does the fusion of individual strategies improve the performance of the individuals query models?
4. Is the results of diversification affected if only the similarity with the previous displayed classified is taken into account?
5. Is the results of diversification affected if the average similarity of previous displayed docs is taken into account?
6. Is the results of diversification affected if only the similarity with the previous four displayed classifieds is taken into account?
7. Does the fusion of diversified systems improve the performance of the not diversified systems?



The remainder of this paper is organized as follows. Second section provides related work. Third section serves as introductory chapter to the of information retrieval and explain important terms of the method that we follow. Section four presents the experimental framework. Fifth section offers experimental results. Section seven analyses the results and eight section concludes this paper and discusses future directions.

Chapter 2


Related Work

 We distinguish between the following **sections of related work** and our contribution: query modeling, data fusion, diversification, data fusion of diversified result lists and **our contribution**. 

2.1 Query modeling

Related work on query representations exists from the early 90's. From the first Text retrieval conferences (TREC) query representation using terms of the topic and routing queries are used [31], [32]. Also from the first TREC conferences it is proven that automatic creation of query representation is as effective as manual.  In 1993, one of the TREC2 experiments was the combination of multiple representations and different treatment of key concepts of a topic like title, description etc. Also, **on** [33] different queries representations are experimented like routing queries, Ad hoc and phrases queries and proved that combination of multiple queries are useful. In 1957, Luhn [34] suggested that automatic text retrieval systems could be designed based on a comparison of content identifiers attached both to the stored texts and to the users information queries. 

2.2 Diversification

Related work in diversification of results started when it was understood that the ranking of the search engines was not enough to cover the information need of different users. In 1998 Carbonell and Gordstein [35] introduced the Maximal Marginal Relevance (MMR) which takes into account the relevance of the document but also the similarity between the other documents. Agrawal et al. on [36], focus on how to diversify search results given ambiguous queries based on the category the results belong. Zhai and Laferty **on** [37] proposed to include some results for each subtopic of the search results. In [38] they proposed to rank the results with a goal to maximize the probability of finding a relevant document among the top n so they can achieve perfect precision using probabilistic model from the Bayesian information retrieved techniques. In [39] they document the different kind of diversification and they created a tool that gives the opportunity to the user to select how to combine relevance with diversity. The choices for diversification of the results is based on context, novelty or different categories. Also, work exists for the measures for the evaluation of diversification **like on** [40]. 



2.3 Data fusion

On the early years of TREC conferences the effectiveness of the result sets fusion was investigated as well. Belkin [on](#) [41], ~~he~~ conducted experiments with combSUM, combMNZ, combANZ, combMIN, combMAX data fusion techniques in two ways. The first was for the combination of query formulations and the second for the combination of two different data collections. They concluded that combining multiple pieces of evidence as query formulations is a beneficial way to increase retrieval effectiveness.

Lee [on](#) [42] influenced by Belkin [gave some insight on the evidence](#) that different runs retrieve similar sets of relevant documents and different sets of non-relevant [docs](#). Also, he evaluated existing data fusion techniques (combSUM, combMNZ, combANZ, combMIN, combMAX) and combGMNZ using different similarity algorithms as well as query formulations. It is proved that CombMNZ [provide](#) better retrieval effectiveness than the others because combMNZ favors documents retrieved by multiple runs. He also identified that in the case of combination of multiple runs, higher relevance overlap than non-relevance overlap on the retrieved set can improve system effectiveness. Lee did not identify the exact difference needed to improve effectiveness. Also, he did not use the most effective result sets available, but rather, selected his test sets at random. Furthermore, he used result sets from entirely different information retrieval systems. This does not simply vary the retrieval strategy used for the experiments, but all retrieval utilities and other systemic differences.

Chowdhury [on](#) [43] investigated the fusion of highly effective retrieval strategies keeping the systemic properties stable. He concluded that it doesn't tend to improve retrieval effectiveness but he used a limited amount of data and query models.

Beitzel [at al](#) [44] experimented with [high](#) effective retrieval strategies as well [to clarify](#) the conditions required to improve effectiveness of data fusion. He concluded that significant number of unique relevant docs is required, not a simple difference between relevant and non-relevant overlap as previously thought. From these results, it is clear that voting is highly detrimental to fusion in the case of fusing highly effective retrieval strategies in the same system. On the other hand, [on](#) [45] they proved the opposite. They keep stable the systemic properties like query modeling, stemming, document presentation [stable etc](#) and they experiment with different highly effective retrieval strategies. Their goal was to prove that the believe that the combination of highly effective retrieval systems is an effective way to fuse result sets. They have shown effectiveness cannot be improved by fusing highly effective retrieval strategies.

Other related work on data fusion can be found on [46] they explained and contacted experiments with three data fusion algorithms (Rank position, Boda count, and Condorcet). The first one takes into account the position of the results and the other two are voting the results. They also contact experiments using the best, bias and all systems.

2.4 Data fusion of diversified result lists

Recently [he](#) first attempt to utilize data fusion for diversification was [on](#) [47]. They proposed their fusion method and they prove that data fusion outperform [the](#) existing diversification methods.



2.5 Our contribution

Our contribution ^{to} ~~on~~ the query modeling topic is that we ~~will~~ compare multiple retrieval systems taking into account different systemic properties like query models, retrieval strategies and preprocessing. Also, we will investigate if an extra weight on documents with attributes will add any value to the retrieval performance.

On fusion of the results, we used compMNZ to fuse results with stable systemic differences but with different retrieval algorithms. The approach is the same as Chowdhury ~~on~~ [43] but with the difference that we will use also the description, attributes and category fields of the classified on the query models instead of only title. This means that we will prove if his conclusion was true that fusion of retrieval strategies doesn't tend to improve retrieval effectiveness.

On ~~D~~iversification of results, we will investigate the impact on performance of comparing a specific document set instead of comparing the similarity of the entire set of document. Also, proposing three alternative diversification methods, we will investigate the impact on diversification of using window on comparing results documents.

Similar to [47] ~~that they~~ proposed to diversify fused results lists, we merge diversified results. To the best of our knowledge, the fusion of diversified results ~~is~~ not investigated yet. So this is the first attempt to see if the performance of diversified results is affected by the data fusion.

Chapter 3

Method

We described the methods used in this thesis in the following parts:

- . Query modeling
- . Retrieval modeling
- . Diversification
- . Late data fusion
- . Data fusion of diversified result lists



3.1 Query modeling

Query modeling is the procedure to create a bag of words or a word to represent the information need. This bag of words or word is the query. Query modeling creation involves the preprocessing step and the identification of tokens. Even if all previous steps are chosen and implemented to improve the accuracy of relevant results, the right query modeling will find highly related documents. The query can be consisted of the parts of a document in the case of existed structure or they can be consisted of phrases or selected terms. We explore three different kind of query models: a) query models created by the fields of the source classified, b) query models consisted of characteristic words of the source classified based on the log likelihood ratio of each word and c) queries created by the information given of an initially returned results (pseudo relevance feedback).

3.1.1 Source classified fields' query models

The source classified that the user just clicked is a great resource that indicates user's information need. Thus, important information can be extracted from the source classified for creating query models. Classifieds typically consisted of title, description and category. Some classifieds consisted also by arguments. Title summarizes the contents of the classified. Description has more details of the classified. Category is the category that they classified is part of. Arguments are depended on the classified. In some cases is the size of a product or the color and it is not required to be filled in. Combination of contents' fields are mapped to queries and multiple query models are created (see table 1).

Table 3.1: Query models' explanation is presented. Explanation is given for the use of fields in the query model, log likelihood ratio(LLR), stemming and pseudo relevance feedback(PRF).

Name	Fields	Stemming	LLR	PRF
T	title	no	no	no
T-LLR	title	no	yes	no
T stemming	title	yes	no	no
T-LLR stemming	title	yes	yes	no
T+D	title, description	no	no	no
T+D+A+C	title, description, category, attributes	no	no	no
A+C	attributes, category	no	no	no
T+D+A+C-LLR	title, description, category, attributes	no	yes	no
T+D+A+C-Pseudo	title, description, category, attributes	no	no	yes
T-Pseudo	title	no	no	yes

3.1.2 Pseudo Relevance Feedback

The general idea behind relevance feedback is to take the results that are initially returned from a given query and to use information about whether or not those results are relevant to perform a new query [12]. Pseudo relevance feedback provides an automated way to have feedback for a query. With the creation of a new query based on this feedback the retrieval performance improves.

Although, pseudo relevance feedback improves the efficiency of the system, it is dependable to the original query since it assumes that the top k results are relevant. However, through a query expansion, some relevant documents missed in the initial round can then be retrieved to improve the overall performance. Clearly, the effect of this method strongly relies on the quality of selected expansion terms.

3.1.3 Log Likelihood Ratio

One of query models is Log Likelihood Ratio(LLR).

“A likelihood ratio test is a statistical test used to compare the fit of two models, one of which (the null model) is a special case of the other (the alternative model). The test is based on the likelihood ratio, which expresses how many times more likely the data are under one model than the other.”

As it is described in [14] we can use LLR to compare ~~corpus~~ ^{corpora} and find the terms of a corpus that are more characteristic. There are two main types of corpus comparison:

- . Comparison of a sample corpus to a larger corpus (normative)
- . Comparison of two (roughly-) equal sized corpora

These two main types of comparison can be extended to the comparison of more than two corpora. For example, we may compare one normative corpus to several smaller corpora at the same time, or compare three or more equal sized corpora to each other. In general, however, this makes the results more difficult to interpret.

This first type of comparison is intended to discover features in the sample corpus with significantly different usage (i.e. frequency) to that found in “general” language. While second type aims to discover features in the corpora that distinguish one from another. In our case, the first type is more appropriate since we need to find a way to distinguish a model for a classified



against a large corpus that will give us enough feedback for every word in the classified. We refer to the larger corpus as a “normative” corpus since it provides a text norm (or standard) against which we can compare.

The representativeness of the big corpus needs to be considered when comparing two corpora. It should contain samples of all major text types and if possible in some way proportional to their usage in the natural writing of a classified in case we want features (in our case frequencies) to make sense. In the case of classifieds created by users, we need a corpora with a data set of classifieds really created by users and big to contain almost all different words a user will write in his classified [14].

We can create query models with the use of LLR using the top k words with the biggest LLR number. This means that we have to calculate LL for every word in a given classified.

The method that we have to follow is the following: Given a visited classified as null corpora and a big dataset of classifieds as normative corpora that we wish to compare with, we produce a frequency list. This would be a word frequency list. For each word in the first frequency list we calculate the log-likelihood statistic. This is performed by constructing a contingency table see table 2.

Table 3.2: Contingency table for Log likelihood calculation.

	First Corpus	Second Corpus	Total
Frequency of word	a	b	a+b
Frequency of other words	c-a	d-b	c+d-a-b
Total	c	d	c+d

Then, we need to calculate the expected values (E) according to the following formula:

$$E_i = \frac{N_i \sum_i O_i}{\sum_i N_i} \quad (3.1)$$

The calculation for the expected values takes into account the size of the two corpora, so we do not need to normalize the figures before applying the formula. We can then calculate the log-likelihood value according to this formula:

$$-2 \ln = 2 \sum_i O_i \ln \frac{O_i}{E_i} \quad (3.2)$$

This equates to calculating log-likelihood LL as follows:

$$LL = 2 * ((a * \ln \frac{a}{E1}) + (b * \ln \frac{b}{E2})) \quad (3.3)$$

The word frequency list is then sorted by the resulting LL values. This gives the effect of placing the largest LL value at the top of the list representing the word which has the most significant relative frequency difference between the two corpora. In this way, we can find the words most indicative (or characteristic) of one corpus, as compared to the other corpus, at the top of the list [14].

3.2 Retrieval modeling

A retrieval model takes a query and a document as input and identifies a measure of relevance between the query and the document. Different retrieval models have different retrieval strategies thus resulted documents differs as well.

Retrieval model or ranking function used by search engines mostly. A search engine except of finding the relevant document, has to rank and order them by relevance. This is typically done by assigning a numerical score to each document based on a ranking function, which incorporates features of the document, the query, and the overall document collection.

The study of retrieval models is central to information retrieval. Many different retrieval models have been proposed and tested, including vector space models, probabilistic models and logic-based model.

The ranking functions-retrieval models that we will use for this project are the following:

- . TfIdf
- . Okapi BM25
- . Probabilistic Language Model

3.2.1 TfIdf

TfIdf (term frequency-inverse document frequency) is a kind of common methods used as term weighting factor in information retrieval. This retrieval method ranks documents based on characteristic terms of a document. Characteristic terms for a document are those who only frequently appears in the possible relevant document while infrequently in the rest documents of data collection [7].

TF is words frequency and idf is inverse document frequency. Term frequency in the given document is the number of times a given term appears in that document. The inverse document frequency is a measure of whether the term is common or rare across all documents.

TfIdf is calculated as:

$$TfIdf = tf * idf \quad (3.4)$$

$$idf = \log \frac{d}{dt} \quad (3.5)$$

Where

d : total number of documents in the collection

dt : total number of documents where term t occurs

However if the term t does not occur in the document collection idf, then dt will be equal to zero. Therefore the formula is adjusted to $1+dt$

TfIdf advantage is that it tends to filter out common terms. When a document has high term frequency while the term appears rarely in the whole collection of documents then it has high weight in TfIdf scoring.

3.2.2 Okapi BM25

In information retrieval, Okapi Best match 25 (BM25) is a ranking function used by search engines to rank matching documents according to their relevance to a given search query [9] . Okapi ranking function is based on the probabilistic retrieval framework. It makes an estimation

of the probability of finding if a document d_j is relevant to a query q . Three factors affects Okapi's score. First is the query terms frequency, second is the inverted frequency of query terms and finally, the length of the document. With this way, it scores higher a short document that mention all query terms.

Given a query , containing keywords , the BM25 score of a document is:

$$BM25(d_j, q_i : N) = \frac{Idf(q_i) * Tf(q_i, d_j) * (k + 1)}{(tf(q_i, d_j) + k * (1 - b + (b * |d_j|/L)))} \quad (3.6)$$

Where N : total number of documents

$tf(q_i, d_j)$: the frequency of q_i word in d_j document

$idf(q_i)$: is the inverse document frequency of word given by:

$$idf(q_i) = \log \frac{N - DF(q_i) + 0.5}{DF(q_i) + 0.5} \quad (3.7)$$

d_j : is the length of document d_j in words L : is the average document length in the corpus

3.2.3 Probabilistic Language Modeling

Language Modeling is the task of estimating the probability distribution of linguistic units such as words, sentences, queries, utterances, or even complete documents. The probability distribution itself is referred to as a language model [10].

Given the query q and the user U , we want to find the most probable documents. That is, we want to rank the documents by $p(d|q, U)$.

Using Bayes' theorem,

$$p(d|q, U) = \frac{p(d|U)p(q|d, U)}{p(q|U)} \quad (3.8)$$

For the purpose of ranking, we can ignore the denominator and define the relevance of a document as:

$$pq(d) = p(d|U) * p(q|d, U) \quad (3.9)$$

The query likelihood $p(q|d)$ is calculated by assuming that the query terms are independent, and then multiplying the probabilities for the individual terms. If the query $q = (q_1 q_2 \dots q_m)$, then:

$$p(q|d) = \prod_{i=1}^m p(q_i|d)$$


Furthermore, suppose that we have the query “This is a great book for retrieval and evaluation in IR” created by the description of a book and also we have as candidate document with description “This is a book for evaluation in IR”. The candidate document does not contain the query word “retrieval”. Now, if we estimate $p(\text{retrieval}|d)$, then this probability will be zero and the query likelihood will vanish. Thus, the language model for a document has to distribute some probability mass among words that are not in the document too. This task is called smoothing [11]. Dirichlet smoothing is used to solve the zero probability and data sparseness problems.

$$p(q|d) = \frac{tf + m * p(q|C)}{|D| + m} \quad (3.10)$$

3.3 Diversification

Maximal marginal relevance

Diversification is implemented to provide of more diversified result set. Maximal Marginal Relevance (MMR) is a diversification method aims to re rank the result set selecting the highest combination of a similarity score with respect to a query and similarity score with respect to the documents selected at earlier rank.



$$MMR \stackrel{def}{=} \underset{D_i \in R \setminus S}{\text{Arg max}} [\lambda (Sim_1(D_i, Q) - (1 - \lambda) \max_{D_j \in S} Sim_2(D_i, D_j))] \quad (3.11)$$

where: R : Rank list of documents retrieved by an IR system

S : is the subset of documents in R already selected

Sim1 : is the similarity between documents and a query

Sim2 : the similarity between the documents



lamda : 0.5 because we want to give the same weight to ranking order and diversity

3.3.1 Maximal marginal relevance alternative 1

Maximal Marginal Relevance alternative (MMRalt) is a diversification method aims to re rank the result set selecting the highest combination of a similarity score with respect to a query and similarity score with respect to the previous selected document at earlier rank.

Algorithm:

1. while we still have documents not selected
 - (a) choose the first one and expose it
 - (b) already displayed list : document0 (document in the first ranked position)
 - (c) calculate the MMR of documentX using as cosine similarity the similarity between X and already displayed
 - (d) expose the documentZ with the max(MMR) as the next one
 - (e) already displayed: documentZ

$$MMRAlt \stackrel{def}{=} \underset{D_i \in R \setminus S}{\text{Arg max}} [\lambda (Sim_1(D_i, Q) - (1 - \lambda) Sim_2(D_i, D_s))] \quad (3.12)$$

R : Rank list of documents retrieved by an IR system

S : is the subset of documents in R already selected

s : is the previous document selected

Sim1 : is the similarity between documents and a query

Sim2 : the similarity between the documents

lamda : 0.5 because we want to give the same weight to ranking order and diversity

3.3.2 Maximal marginal relevance alternative 2

Maximal Marginal Relevance alternative 2 (MMRalt2) is a diversification method aims to re rank the result set selecting the highest combination of a similarity score with respect to a query and similarity score with respect to the previous selected documents at earlier rank. The difference between MMR and MMRalt2 is that this technique it does not take into account the maximum cosine similarity between X document and the displayed documents set. Instead it takes into account the average MMRalt2 score between a document and the previous displayed documents.

Algorithm:

1. while we still have documents not selected
 - (a) choose the first one and expose it
 - (b) already displayed list : document0 (document in the first ranked position)
 - (c) $MMR_x = \text{AVG of } MMR_{x,ds}$ where ds set of the displayed documents
 - (d) expose the documentZ with the $\max(MMR_x)$ as the next one
 - (e) already displayed: list with document0, documentZ

$$MMRAlt2 \stackrel{def}{=} \underset{D_i \in R, s}{\text{Arg max}} \quad (3.13)$$

R :Rank list of documents retrieved by an IR system

S :is the previous document selected

3.3.3 Maximal marginal relevance average last four

Maximal Marginal Relevance average last four (MMRalt2last4) is a diversification method aims to re rank the result set selecting the highest combination of a similarity score with respect to a query and similarity score with respect to the previous four selected documents at earlier rank. The difference between MMRalt2 and this technique is that it takes into account only last four selected documents instead of all of them.

The selection of the window of four is based on the graphics on

Algorithm:

1. while we still have documents not selected
 - (a) choose the first one and expose it
 - (b) already displayed list : document0 (document in the first ranked position)
 - (c) $MMR_x = \text{AVG of } MMR_{x,ds}$ where ds set of the four last displayed documents
 - (d) expose the documentZ with the $\max(MMR_x)$ as the next one
 - (e) already displayed: list with document0, documentZ

3.4 Late Data Fusion

Based on [15], data fusion is the process of integration of multiple data and knowledge representing the same real-world object into a consistent, accurate, and useful representation. There are two approaches for the combination of data known as early fusion or late fusion. Early fusion is the combination of data prior to indexing. Thus the data aggregated and then retrieval model use aggregated data as input. While late fusion assumes each source of data has associated with it some form of a ranking function, each of which can be independently queried. Once each source has been queried, the outputs of each of these queries can be aggregated together to form a final response to the initial query [16].

Since different retrieval results can generate quite different ranges of similarity values, a normalization method should be applied to each retrieval result. Normalization controls the ranges of similarity values that retrieval systems generate. Hence, in order to align both the lower bounds of similarity values and the upper, we normalize each similarity value by the maximum and minimum actually seen in a retrieval result as follows:

$$normalizedscore(x) = \frac{x - min}{max - min} \quad (3.14)$$

where



min : the minimum value for the ranked list

max : the maximum value for the ranked list

After the normalization of the score we merged all of the ranked lists. We consider the following late data fusion methods:

Table 3.3: Late data fusion methods explanation.

Name	Explanation
compMAX	Maximum of individual scores
combMIN	Minimum of individual scores
combSUM	Sum of individual scores
combANZ	combSUM / number of non zero scores
combMNZ	combSUM * number of non zero scores
WcombSUM	weighted sum of individual scores
WcombMNZ	WcombSUM * count of non zero results
WcompWW	WcombSUM * sum of individual weights



3.5 Data fusion of diversified result lists

All of the merged ranked lists are diversified with the diversification methods described in a previous section.

Chapter 4

Experiment Design



To answer the the research questions, we contacted six different kind of experiments. In the following sections the different types of experiments, data, research questions and evaluation are presented.

4.1 Data and data gathering

We use as visited classifieds a dataset of 100 classifieds and our data collection is a dataset of 7.502.132 classifieds (8.8 GB) which is the total number of active classifieds. Also, we remove classifieds that are suspended from the system due to duplicates that we found in the preliminary evaluation.

The classifieds were uniformly formatted into an Standard Generalized Markup Language(SGML) structure with tags for each part of a classified, as can be seen in the following example.

Listing 4.1: SGML formatted classified

```
<DOC>
  <DOCNO>244563422</DOCNO>
  <TITLE>
    koop huur rietgedekte villa landhuis praktijkruimte eerbeek
  </TITLE>
  <DESCRIPTION>
    aangeboden exclusief rietgedekt modern landhuis eerbeek
    praktijkruimte eigen ingang koopprijs 998 000 kk
    huurprijs 3 300 per maand
  </DESCRIPTION>
  <CATEGORY>
    huizen en kamers huizen te koop huizen koop
  </CATEGORY>
  <PRICE>
    998.000
  </PRICE>
  <ATTRIBUTES>
    Aantal kamers 5 kamers of meer Woonoppervlakte 150 m of
    meer
  </ATTRIBUTES>
</DOC>
```

All classifieds had beginning and end markers, and unique DOCNO id field. Also, they are consisted of a title, a description, a price, a category and several attributes. Attributes are not the same in each classified.

With the use of Indri build Index application we build repositories from the document collection. The buildIndex application uses parameter files to create repositories of indexes of all the documents (see listing 2).

Listing 4.2: Build index parameter file

```
<parameters>
  <index>/home/varvara/workspace/externalSources/indri/
    repositories2/mergedOutput24</index>
  <memory>1G</memory>
  <corpus>
    <path>/home/lemur/testdata/firstCorpus</path>
    <class>trectext</class>
  </corpus>
  <stemmer><name>krovetz</name></stemmer>
  <field>
    <name>p</name>
  </field>
</parameters>
```



Then, we merged the 125 repositories in six repositories with the use of dumpIndex application. Statistics for each repository of unstemmed data collection you can find in table 3 and stemmed in table 4.

Table 4.1: Unstemmed repositories statistics of five repositories (Rep1, Rep2, Rep3, Rep4 and Rep5). Total number of documents, unique terms and total terms for each repository is given.

	Rep1	Rep2	Rep3	Rep4	Rep5
Documents	1440000	1403211	1440000	1440000	1440000
Unique terms	1126145	1138034	1135226	1057184	1066359
Total Terms	76223460	67441399	68602942	68034636	68206373

Table 4.2: Stemmed repositories statistics of five repositories (Rep1, Rep2, Rep3, Rep4 and Rep5). Total number of documents, unique terms and total terms for each repository is given.

	Rep1	Rep2	Rep3	Rep4	Rep5
Documents	1440000	1403211	1440000	1440000	1440000
Unique terms	1074828	1081013	1079051	1006485	1016282
Total Terms	76184789	67405711	68567148	68002150	68173834

4.2 Results Retrieval

Having indexes of the document collection described above, the next step is to create an Indri-style query file like listing 3.

Listing 4.3: Query parameter file

```
<parameters>
<index>/home/repositories/rep1</index>
<query>
<text> koop huur rietgedekte villa </text>
<number> 244 </number>
</query>
<baseline>tf.idf,k1:1.0,b:0.3</baseline>
<count>30</count>
<trecFormat>true</trecFormat>
</parameters>
```

We use 100 classifieds as a sample of users last visited classified. We create queries for each of the 100 visited classifieds in Indri-style query files like in listing 3. Also, creating the query involves parsing the visited classified which we aim to find relevant classifieds. Parsing has to be the same with the preprocessing of the document collection. Otherwise the accuracy of results will be affected negatively and the results will be not the desired. For example if we have a query term 'books' then it will be difficult to find documents relative to 'book'.

Next, the query file runs against our repositories using IndriRunQuery and retrieves the relevant list of classifieds. The output of the IndriRunQuery is a TREC style qrels file. These files are then input to trec_eval (see section 5.1), to calculate the evaluation metrics.

The different query models we use are the following:

- . Title words
- . Title and description words (unstructured data)
- . Attributes and Category (structure data)
- . Structure data and unstructured data
- . LLR in Title words
- . LLR in Title and description words
- . LLR in Structure data and in unstructured data

Different retrieval systems are used for experimentation purposes. As it is mentioned in section 3.4 , our three retrieval models are:

- . Tf.Idf
- . Okapi BM25
- . LM

4.3 Our Baseline

The title of the classified is the summary of the classified. It's brief and consisted of the most important information. For this reason, we use the title query model as baseline with okapi as retrieval strategy without stemming. We choose okapi that it is proven that is a state of the art retrieval strategy and it performs better on our experiments in a comparison with the other two. We are not using stemming because it harms our retrieval effectiveness and this is proven from our experiments as well.

4.4 Experiments

4.4.1 Stemming experiment-Experiment



We make one initial experiment to measure if any change on performance occurs when we use stemming in the preprocessing phase. In this phase we use two different query models and we compare the result lists with stemming used in the preprocessing and without.

Preprocessing is a good approach to improve the effectiveness of retrieval systems. The document collection consists of classifieds created by regular users and contains a lot of noise which should be extracted before indexing takes place. Removing the noise can improve query efficiency.

We preprocess the document collection and we convert it in lowercase, remove stopwords, replace punctuations with spaces and remove words with one or two characters. Also, for experimentation purposes we compare two approaches (with or without stemming). Stemming is based on the snowball stemmer [20]. Then, the data are saved in SGML formatted documents.

The preprocessing approaches that we are comparing are the following:

Approach 1-No stemming:

1. Parsing
2. Remove Stop words
3. Storing documents on Standard Generalized Markup Language format

Second Approach-Stemming:



1. Parsing
2. Remove Stop words
3. Stemming
4. Storing documents on Standard Generalized Markup Language format

Table 4.3: Systems used on the stemming experiment are presented. Explanation is given for the use of fields use in the query model, log likelihood ratio(LLR), stemming and retrieval strategy.

Name	Fields	Retrieval strategy	Stemming	LLR
T-TfIdf	title	TfIdf	No	No
T-LM	title	LM	No	No
T-Okapi	title	Okapi	No	No
T-TfIdf-LLR	title	TfIdf	Yes	No
T-LM-LLR	title	LM	Yes	No
T-Okapi-LLR	title	Okapi	Yes	No
T-TfIdf-stemming	title	TfIdf	No	Yes
T-LM-stemming	title	LM	No	Yes
T-Okapi-stemming	title	Okapi	No	Yes
T-TfIdf-stemming-LLR	title	TfIdf	Yes	Yes
T-LM-stemming-LLR	title	LM	Yes	Yes
T-Okapi-stemming-LLR	title	Okapi	Yes	Yes

After we compare the results of this experiment, we decide that we will not use stemming in the rest of the query models.

4.4.2 Query modeling experiments

With this kind of experiment, we answer the first research question. For each of the three types of query models (classifieds' structure, discriminative terms, pseudo relevance feedback) we construct queries and submit them to our index of classifieds. We measure the performance of each system and we compare the results to evaluate the differences in performance between the models.

To create a good query model, the query has to contain the most important information of the classified. But how can we find words that contribute the most important information from a classified? In the case of classifieds we have several parts that important information can be found. Also, a lot of noise exists in the fields which harms the retrieval efficiency. Discriminative terms can be extracted by the classifieds fields. Furthermore, we can get feedback from the result lists and create new queries to improve our retrieval performance.

As a first step, we conducted an experiment to answer the first research question using query models with classifieds fields terms. We compare the result lists with our baseline to find if any of the fields affects the performance of the baselines retrieval system. In this query models, the query is the preprocessed classifieds field either alone or combined. The fields that are used are the following:

1. Title
2. Description
3. Category
4. Attributes

Table 4.4: Individual systems used on the query model experiment are presented. Explanation is given for the use of fields use in the query model, log likelihood ratio(LLR), stemming and pseudo relevance feedback(PRF).

Name	Fields	Retrieval strategy	Stemming	LLR	PRF
T-TfIdf	title	TfIdf	No	No	No
T-LM	title	LM	No	No	No
T-Okapi	title	Okapi	No	No	No
(T+D) TfIdf	title, description	TfIdf	No	No	No
(T+D) LM	title, description	LM	No	No	No
(T+D) Okapi	title, description	Okapi	No	No	No
(T+D+A+C) TfIdf	title, description, category, attributes	TfIdf	No	No	No
(T+D+A+C) LM	title, description, category, attributes	LM	No	No	No
(T+D+A+C) Okapi	title, description, category, attributes	Okapi	No	No	No
(A+C) TfIdf	category, attributes	TfIdf	No	No	No
(A+C) LM	category, attributes	LM	No	No	No
(A+C) Okapi	category, attributes	Okapi	No	No	No

We implement some more query models used in the rest of the experiments using discriminative terms and pseudo relevance feedback.

Query models with discriminative terms

As it is described in the methodology, we will use the LLR to create queries with discriminative terms. We used LLR to extract discriminative terms from the title and **in** the entire classified. Given the visited classified's field or all fields as null corpora and the big dataset of 8.8 GB classifieds as normative corpora we will produce our **the** queries.

Query models with pseudo relevance feedback

The method we follow in order to use pseudo relevance feedback is the same method used in normal retrieval. The system will use the results from the original query and extend it with the feedback. The system assumes that the top five ranked documents are relevant. It extracts the five most frequent terms in this top five ads, expands the query with this terms and finally retrieves results with the expanded query.

Table 4.5: Query models' explanation is presented. Explanation is given for the use of fields use in the query model, log likelihood ratio(LLR), stemming and pseudo relevance feedback(PRF).

Name	Fields	Stemming	LLR	PRF
T	title	No	No	No
T-LLR	title	Yes	No	No
T-stemming	title	No	Yes	No
T-stemming-LLR	title	Yes	Yes	No
T+D	title, description	No	No	No
T+D+A+C	title, description, category, attributes	No	No	No
A+C	category, attributes	No	No	No
T+D+A+C-LLR	title, description, category, attributes	Yes	No	No
T+D+A+C-Pseudo	title, description, category, attributes	No	No	Yes
T-Pseudo	title	No	No	Yes

4.4.3 Retrieval methods

We **contacted** this experiment to investigate which of the retrieval methods is performing better(second research question). All the query models available are part of the experiment. The three retrieval methods are already mentioned in the previous chapter.

4.4.4 Late Data Fusion experiments

With the late data fusion experiments, we give answers to the third research question. We use late fusion techniques to fuse the individual models from the query modeling experiments and we produce new result lists. Then we compare their performance in contrast with individual models.

Late fusion techniques are applied to individuals ranked lists to answer the third research question. We experiment with eight fusion techniques as explained in methodology chapter. Results compared with the individual models to answer the relevant research question. Since we don't have any weight in the queries, we use the MAP as a weight of a sample of classifieds runs. So we choose 50 random visited classifieds and we use the MAP of the trec results run. Then, we use the MAP as the weight to the WCombSUM, WcombMNZ and WcombWW methods.

Table 4.6: Late Data Fusion(LDF) methods used on the experiments

Name
combMAX
combMIN
combSUM
combMNZ
combANZ
WcombSUM
WcombMNZ
WcombWW

4.4.5 Diversification experiments

To answer the research question four to six, we diversify first the query models from the first experiment with MMR method. Then we diversify with our three alternatives diversity methods explained on method section. To evaluate the results, we compare the alternatives with the existed MMR method.

To evaluate these experiments we used the clicks logs evaluation method described below. The assessors evaluation is based one the assumption that we are searching relevant classifieds to the visited one. While in the case of diversification, we want more diversified results in case that we will increase the possibilities that will cover the information need of the user. Clicks indicate user's interest.

MMRalt1

The algorithm of this alternative diversification is provided on methodology chapter. It is based on the assumption that we don't want to show two similar results in a row. So, we are taking into account only the similarity between the previous selected classified and the unselected classifieds.

MMRalt2

The algorithm of this alternative diversification is provided on methodology chapter. The difference with the MMR implementation is that we are calculating the avg similarity for each classified with all the selected classifieds.

MMRaltAvgLst4

To answer the sixth research question we implement the algorithm provided **on** methodology and we compared the results with MMR.

Since we were doing the experiments in an industrial environment, their decisions or experience affect our decision in some cases. They wanted to expose only five results as similar classifieds paginated. This creates the idea to use windows on comparison of similarity. So the basic idea is that we want to show 5 diversified results per page. We compare only the similarity of the not selected classifieds and the previous four displayed classifieds. The algorithm we use to implement is provided on methodology chapter.

4.4.6 Data fusion of diversified result experiment

For the last experiment, using the same late fusion techniques as in the previous experiment, we merge the diversified result lists to answer the seventh research question. We measure the difference in performance comparing the new result lists with the fused results from the previous experiments. The evaluation of the results is based on the click log evaluation as well due to the reasons mentioned in the previous section.

4.5 Evaluation



To evaluate if a system is performing better than **the rest**, we need to know which documents are relevant and if they are retrieved by the specific system. We follow two ways of evaluation: a) assessors evaluation b) click logs evaluation

4.5.1 Assessors Evaluation



Our evaluation will be based on Text REtrieval Conference (TREC). Three assessors are provided with 100 users' information needs which in our case **is** the contents of the visited classifieds and they evaluate a ranked list of documents (results of each system). They judge documents as relevant or not.

Of the three components of a test collection - the document set, a set of information need statements called topics, and the relevance judgments that indicate which documents should be retrieved in response to a given topic - the relevance judgments are the most expensive to produce [23]. Within big document collections, judging all documents as relevant or not is almost impossible due to the time it requires. Also based on [27] the greater the ranked position of a relevant document (of any relevance level) the less valuable it is for the user, because the less likely it is that the user will examine the document. It would therefore be desirable from the assessor viewpoint to rank highly relevant documents highest in the retrieval results.

With the use of pooling we can judge only a subset of the retrievals output. In pooling, a set of documents to be judged for a topic (the "pool") is constructed by taking the union of the top documents retrieved for the topic by a variety of different retrieval methods. Each document in the pool for a topic is judged for relevance, and documents not in the pool are assumed to be irrelevant to that topic. **The choice of pool depth is ours.** Sakai and Mitamura [25] report the outcome of their experiment to investigate the effect of reducing both the topic set size and the pool depth and they prove that using 100 topics with depth-30 pools generally yields fewer errors than using 30 topics with depth-100 pools.



Due to the fact that different persons have different opinions about the relevance for the same document **we will use multiple assessors.** However, **multiple assessors make errors** which



affect the assessment. Basically the reasons lie in the ambiguity of data or mistakes of annotators due to lack of motivation or knowledge. Also, non-expert assessors judging domain-specific queries make significant errors affecting system evaluation. When assessors are not closely managed or highly trained, mistakes must be common [29]. For this reason we calculate the kappa coefficient (k) to check the reliability of judgments. The kappa coefficient (K) measures pairwise agreement among a set of assessors making binary judgments, correcting for expected chance agreement: $K = \frac{P(A) - P(E)}{1 - P(E)}$ where $P(A)$ is the proportion of times that the assessors agree and $P(E)$ is the proportion of times that we would expect them to agree by chance, calculated along the lines of the intuitive argument presented above [30].

In the following table we present the annotation agreement between our assessors:

Table 4.7: Inter annotator agreement between different assessors (developer a, developer b, business analyst and product owner). Proportion of times assessor agree on relevance ($P(\text{agree-rel})$), proportion of times assessor agree on irrelevance ($P(\text{agree-irr})$), dataset and k-measure.

Assessor a	Assessor b	DataSet	$P(\text{agree-rel})$	$P(\text{agree-irr})$	k-measure
Developer a	Business analyst	1261	0,13	0,39	0,64
Developer a	Product owner	2148	0,25	0,24	0,57
Developer a	Developer b	574	0,22	0,27	0,57
Business analyst	Product owner	820	0,23	0,26	0,54
Developer b	Product owner	574	0,15	0,37	0,52

With the previous in mind, we take the following decisions:

- . To evaluate if a document is relevant or not we need the opinion of ~~possible~~ ^{potential} users of our system.
- . Assessors have to be Dutch speakers. Since my document collection is in Dutch, assessors must be native Dutch speakers as well. The understanding of the language has to be appropriate to understand entirely the contents of the document.
- . Experts with different background in order to cover different kind of users.
- . Assessors without any intentions for the project. We need pure answers without any intention for the project.
- . Binary value for judgment: zero for irrelevant and one for relevant documents.
- . Assessor will see a list of relevant documents. However, this list will be unordered because we ~~don't~~ want to direct assessor's opinion about relevance.
- . Finally, pool depth is five per system since users need to see only a few highly relevant results. Thus, with the judgment of top five will cover all documents that we need to know if they are relevant or not.

As described above, assessors judge a list of classifieds which are the output of several systems. They assign a binary value in each document from the list based on its relevance with the original topic of the visited classified. Zero for irrelevant and one for relevant.

4.5.2 Clicks Evaluation

In this approach, we use click logs as indication of relevance instead of the assessors judgment. Provided with click logs of four days, we create a relevant list of all the classifieds a user visited in one session after the visited classified. We count as relevant only the classifieds which five different users click on ~~them~~.

4.5.3 Measures

TREC is an annual conference started in 1992 co-sponsored and masterminded by the US National Institute of Standards and Technology (NIST), but tracks are largely organized by the participant research groups. It has contributed on many advances in information retrieval techniques as ranking algorithms, improving old ideas and encouraging new ideas and experimentation.

TREC_EVAL is a tool designed for evaluation of various information retrieval systems. It handles collection of documents, queries, and relevance judgments. It takes two documents as input and it calculates various measures for retrieval system evaluation. The measure we are interested in is the precision at first five documents which is more important for similar classifieds.

Chapter 5

Results

5.1 Query modeling

To answer the first research question about the best query model to improve the performance of our baseline, we conduct the query modeling experiment as it is described on experimental design chapter. Following are the results of this experiment.

Table 5.1: System Performance (P@5) of title (T), title and description (T+D), all the fields (T+D+A+C) and attribute and categories (A+C) query models using three retrieval strategies (BM25, LM, Tfidf) and two types of ground truth (editorial and click logs).

	Editorial			Click logs		
	BM25	LM	Tfidf	BM25	LM	Tfidf
T	0.6560	0.5980	0.626	0.1680	0.1660	0.1540
T+D	0.6660	0.5880	0.6500	0.1700	0.1680	0.1600
T+D+A+C	0.7300	0.6460	0.6860	0.1720	0.1700	0.1580
A+C	0.4800	0.3400	0.4500	0.0920	0.0900	0.0600

The table represents the precision at five first results from four different query models. First, we are adding fields to our baseline to see if any improvement occurs. The last query model with only attributes and category is an extra query model to compare with and find extra refinements.

As you can see in the results with the editorial evaluation, adding the description to our baseline shows an improvement to BM25 and Tfidf, but LM. Adding the attributes and category gives a boost to all retrieval strategies. However, precision of query model with attributes and category is lower even of our baseline.

The results using the click logs evaluation indicate a slight increase in the precision when the description is used in the query models. Also, adding the attributes and category, shows a small increase but not on Tfidf. However, the attributes and category alone in the query model are not performing better of our baseline.

The results verify our initial assumption that adding extra information on the title query model can improve the performance. We can give an initial answer before further analysis to the first research question that using all the fields on the query model indicates the highest precision thus is the best one of the available query models.

5.2 Retrieval method

Second research question that we are answering is about the best retrieval method from the three available. In the table ??, we are presenting the results of the experiment we conducted.

Table 5.2: System Performance (P@5) of the retrieval strategies Okapi BM25(BM25), TfIdf, LM and title (T), title and description (T+D), all the fields (T+D+A+C) and attribute and categories (A+C) query models two types of ground truth (editorial and click logs).

	Editorial				Click logs			
	T	T+D	T+D+A+C	A+C	T	T+D	T+D+A+C	A+C
BM25	0.6560	0.6660	0.7300	0.4800	0.1680	0.1700	0.1720	0.0920
LM	0.5980	0.5880	0.6460	0.3400	0.1660	0.1680	0.1700	0.0900
TfIdf	0.626	0.6500	0.6860	0.4500	0.1540	0.1600	0.1700	0.0600



The table represents the precision at five **first** results of the three retrieval strategies using four different query models. As you can see on the results with editorial evaluation, LM has the smallest precision in a comparison with the other two. BM25 is performing better in both editorial and click logs evaluation. Furthermore, in results with click logs evaluation in one query model LM and TfIdf have the same precision. However, in attributes and category query models TfIdf has lower precision than LM.

Provided with the previous results, we can verify that Okapi BM25 is the retrieval strategy which performs better than the other two.

5.3 LDF

Different late data fusion techniques used to answer the third research question about improving the performance of the individual systems using late these techniques.

Table 5.3: System Performance (P@5) of late data fusion methods (combMAX, combMIN, combSUM, combMNZ, combANZ, WcombSUM, WcombMNZ, WcombWW) and the best individual system using two types of ground truth (editorial and click logs).

	Editorial	Clicks logs
combMAX	0.5400	0.1280
combMIN	0.0760	0.0020
combSUM	0.6160	0.1720
combMNZ	0.6460	0.1720
combANZ	0.0800	0.0020
WcombSUM	0.6140	0.1740
WcombMNZ	0.6360	0.1740
WcombWW	0.6320	0.1740
Best Individual	0.7300	0.1720

The table represents all the late fusion methods we use to answer the third research question in a comparison with the individual best system. The most important observation of this table

is that none of the fused systems is performing better than the individual one using the editorial evaluation.

On the results with the editorial evaluation, the compMNZ is performing better than the rest of the fused systems. The rest of the fused systems have small differences in the precision except of the combMIN and combANZ that have very low precision.

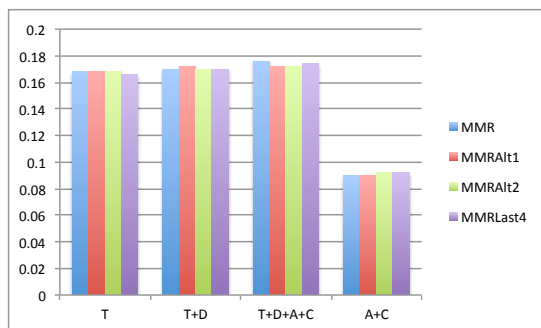
On the results with click logs evaluation, the weighted fused systems have better precision than the rest. Then, a slight decrease in precision is seen on combMNZ and combMAX and a bigger decrease on combMAX. Same as editorial evaluation, the combANZ and combMIN have the lowest precision.

5.4 Diversification

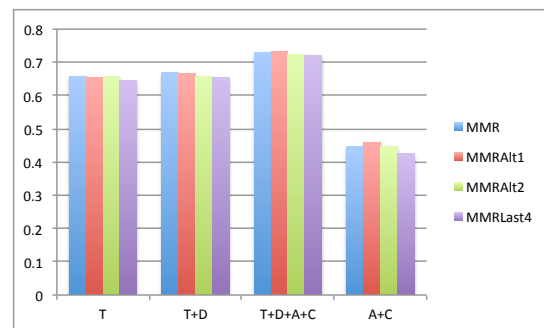
Performance of the different diversification approaches we use on the diversification experiment are presented to graph ??? and ?? to answer the following research questions:

The research questions we aim to answer are the following:

1. **Is** the results of diversification affected if only the similarity with the previous displayed classified is taken into account?
2. **Is** the results of diversification affected if the average similarity of previous displayed docs is taken into account?
3. **Is** the results of diversification affected if only the similarity with the previous four displayed classifieds is taken into account?



(a) Click logs Evaluation



(b) Editorial Evaluation

Figure 5.1: Bar graphs with the system performance(P@5) of diversification methods MMR, MMRAIt1, MMRAIt2, MMRLast4 using title (T), title and description (T+D), all the fields (T+D+A+C) and attribute and categories (A+C) query models and two types of ground truth (editorial and click logs).

In the previous graphs, the precision in five first results for the alternative diversification methods using four different query models is presented. In both graphs the trends are almost the same thus none of the systems performs better than the others. So, as a first and fast answer in the diversification research questions (four, five and six) is that none of them performs better than the MMR method proposed on [35].

5.5 Fused diversified results

Final research question on the improvement fusion of diversified systems improve the performance of the not diversified systems?

Table 5.4: System Performance (P@5) of late data fusion methods (combMAX, combMIN, combSUM, combMNZ, combANZ, WcombSUM, WcomMNZ, WcombWW) of fused system and fused diversified system using two types of ground truth (editorial and click logs).

	Editorial		Click logs	
	Fused diversified	Fused not diversified	Fused diversified	Fused not diversified
combMAX	0.7320	0.5400	0.1700	0.1280
combMIN	0.0800	0.0760	0.0020	0.0020
combSUM	0.6700	0.6160	0.1720	0.1720
combMNZ	0.6700	0.6460	0.1720	0.1720
combANZ	0.0940	0.0800	0.0040	0.0020
WcombSUM	0.6620	0.6140	0.1720	0.1740
WcombMNZ	0.6560	0.6360	0.1720	0.1740
WcompWW	0.6560	0.6320	0.1720	0.1740

The previous two tables represent the comparison of the results of fused diversified systems versus the fusion of individual systems with both click logs and editorial evaluation.

Using the click logs evaluation, we can't see a big difference except of the 0.5 increase on combMAX when the system is diversified.

Using the relevance feedback, we can see a small improvement in precision in all the systems. Also, the biggest difference is in the combMAX which has an increase of 0.19. The smallest increase we have in this table is the combANZ which is 0.014.

All in all, using the editorial evaluation we can verify and answer to the seventh research question that the fused diversified systems perform better than the fused not diversified systems. Using the click log evaluation, we can not give the same answer cause the results in the table doesn't show any big improvement.

Chapter 6

Analysis

6.1 Stemming experiment

As we already mentioned on the experimental design, we first conduct the stemming experiment to see if we will use it in the rest of the experiments or not.

We hypothesize that using stemming will retrieve more relevant results. For example, “cars” will be stemmed in “car” thus if we have a document related to “car” it will be retrieved as well.

Table 6.1: System Performance (P@5) of title (T), title using LLR (T-LLR) query models with and without stemming using three retrieval strategies (BM25, LM, TfIdf) and two types of ground truth (editorial and click logs).

	Editorial			Click logs		
	BM25	LM	TfIdf	BM25	LM	TfIdf
T	0.6560	0.5980	0.626	0.1680	0.1660	0.1540
T-stemming	0.6040	0.5660	0.5640	0.1640	0.1660	0.1600
T-LLR	0.5660	0.5160	0.5280	0.1580	0.1540	0.1500
T-stemming-LLR	0.5280	0.5160	0.4980	0.1580	0.1560	0.1460

Based on the results using editorial evaluation, stemming decrease the P@5 measure at least 0.3 and the maximum of 0.5 at Okapi BM25. In the query model with LLR on title field with LM retrieval strategy, there is no difference in precision. Based on the results using click logs evaluation, there is no big decrease in the results with stemming but neither an improvement. The only case that a small increase in performance happened is on the title query with TFIDF.

We believe that the reason of the previous results is that stemming negatively affects a queries accuracy. For instance, if we search for “organization” and our stemmer removes the reasonably common suffix “ization” we end up with classifieds about “organ”. Also, we found examples that didn’t retrieve any relevant classifieds on the first five results, while there were at least two relevant results in the systems without stemming. Also, we observe that the same classifieds in the result list have lower score due to the fact that the frequency of stemmed terms are greater.

For a further analysis, we are presenting the following table with the count of relevant results retrieved only when stemming used and the count of relevant results retrieved only in

the systems without stemming. We used the Okapi Title query model since we found the biggest difference in the performance.

Table 6.2: Number of relevant classifieds retrieved on the top five results, number of examples retrieve uniquely this number of classifieds with title query model (T) and BM25 without stemming, number of examples retrieve uniquely this number of classifieds with title query model and BM25 with stemming (T-stemming).

	Relevant classifieds	T	T-stemming
1		31	26
2		12	5
3		3	2
4		1	0

As you can see, the uniquely retrieved results from the non stemmed system are more than the uniquely retrieved by the stemmed system. There are 31 visited classifieds retrieved one unique relevant classified that is not retrieved in the first 5 results of the stemmed systems. Also, there are 16 visited classifieds retrieved more than one unique relevant classified that is not retrieved in the first five results of the stemmed systems.

All in all, none of the systems using stemming improve the performance of our baseline thus we decide to **don't** use it in the rest of the experiments.

6.2 Query modeling

We hypothesize that if we add information in our baseline, we will improve the precision of the five first results. In the results provided in the results chapter, this assumption is proved. Adding the description in the baseline's query model improved the performance. Furthermore, adding the attributes and category had the best precision in all three retrieval strategies using either editorial evaluation or click log evaluation.

Also, in the moment we made the editorial evaluation we observed the following:

- . In some classifieds like men/women shoes, there is no difference in the text of the classifieds except of the category name. Thus, queries like title and description do not perform as well as query models including the category name.
- . Also in the same case, attributes will add extra value due to the extra information about the size and the kind of shoe.
- . In the case of products sale classifieds as original classified, we have as result classifieds with parts for this product. However, this is not always relevant.

To further analyze the results we are presenting the following table with the extra relevant results retrieved when description is added on the query model.

As you can see, the query model with title and description has a lot of relevant results retrieved while they are not retrieved from title query model. Although there are much more cases that only one relevant result retrieved uniquely from only title query (26 versus 15), the

Table 6.3: Number of relevant classifieds retrieved on the top five results, number of examples retrieve uniquely this number of classifieds with title and description query model and BM25 (T+D), number of examples retrieve uniquely this number of classifieds with title query model and BM25 (T).

Relevant classifieds	T+D	T
1	15	26
2	12	11
3	13	3
4	4	1

number of the rest of the cases (that more than one uniquely retrieved from the description and title query model) outperform it e.g 13 versus 3. In total 44 relevant classifieds uniquely retrieved from title and description query model and 41 from title query model.

Furthermore, we are presenting the following table with the uniquely relevant results retrieved when all the fields are in the query model in a comparison with the description query model.

Table 6.4: Number of relevant classifieds retrieved on the top five results, number of examples retrieve uniquely this number of classifieds with all fields in query model and BM25 (T+D+A+C), number of examples retrieve uniquely this number of classifieds with title and description query model and BM25 (T+D).

Relevant classifieds	T+D+A+C	T+D
1	39	28
2	18	18
3	8	5
4	6	3
5	1	1

In all the rows of the table, the uniquely retrieved results from all the fields are more or equal than description query model. In total, 72 uniquely retrieved results from all the fields query model and 55 from title and description query model. Thus, the extra information on the attributes and category improves the results.

Also, we are presenting the following table with the uniquely relevant results retrieved when all the fields are in the query model in a comparison with the baseline.

In all the rows of the table, the uniquely retrieved results from all the fields are more or equal than our baseline. In total, 75 uniquely retrieved results from all the fields query model and 55 from title query model.

We also observed that query model with attributes and category indicates the lowest precision in a comparison with the rest of the query models. We hypothesize that not enough information exists to attributes and category, thus the results will be negative. To further prove that assumption, we are presenting the following table with the uniquely relevant results re-

Table 6.5: Number of relevant classifieds retrieved on the top five results, number of examples retrieve uniquely this number of classifieds with all fields in query model and BM25 (T+D+A+C), number of examples retrieve uniquely this number of classifieds with title query model and BM25 (T).

Relevant classifieds	T+D+A+C	T
1	37	22
2	16	19
3	12	8
4	9	5
5	1	1

trieved when attributes and category are only in the query model in a comparison with the baseline.

Table 6.6: Number of relevant classifieds retrieved on the top five results, number of examples retrieve uniquely this number of classifieds with attributes and category in query model and BM25 (A+C), number of examples retrieve uniquely this number of classifieds with title query model and BM25 (T).

Relevant classifieds	A+C	T
1	16	17
2	12	16
3	16	18
4	16	29
5	5	9

It is obvious in the previous results that query model with only attribute and category has less uniquely retrieved results than our baseline. Also, in the results chapter the query model using only attributes and category has the lowest precision, thus we can conclude that attributes and category alone are not containing enough information to improve the precision of query with title.

Query models with discriminative terms Our assumption was that discriminative terms will retrieve less results but more accurate due to the fact that a more accurate query is created. Thus, the precision at first five will be increased. To prove this assumption, we present the following table with the six different systems.

Table 6.7: System Performance (P@5) of title (T), title using LLR (T-LLR), title using stemming (T-stemming), title using both LLR and stemming (T-LLR stemming), all fields (T+D+A+C) and all fields using LLR (T+D+A+C-LLR) query models and two types of ground truth (editorial and click logs).

	Editorial			Click logs		
	BM25	LM	TfIdf	BM25	LM	TfIdf
T	0.6560	0.5980	0.626	0.1680	0.1660	0.1540
T-LLR	0.5660	0.5160	0.5280	0.1580	0.1540	0.1500
T-stemming	0.6040	0.5660	0.5640	0.1640	0.1660	0.1600
T-stemming-LLR	0.5280	0.5160	0.4980	0.1580	0.1560	0.1460
T+D+A+C	0.7300	0.6460	0.6860	0.1720	0.1700	0.1580
T+D+A+C-LLR	0.6360	0.5400	0.5740	0.1600	0.1640	0.1540

Based on the results, all the systems are negatively affected in a comparison with the systems with the discriminative terms. Using the editorial evaluation, big decrease from all the fields query model to LLR all the fields query model in all three retrieval strategies is indicated. Smaller decreases in the rest of the systems are shown but none of them show any improvement. Since, using stemming and relevance in the title query hurts the performance even more than 0.10, we decided to **don't** use it in the rest of the query models.

We hypothesize that the reason of the previous results is that LLR is removing some important information thus relevant results are not retrieved. To prove that, we are presenting the following table with the count of relevant results retrieved only when LLR is used in a system and the count of relevant results retrieved only in the same system without the use of LLR.

Table 6.8: Number of relevant classifieds retrieved on the top five results, number of examples retrieve uniquely this number of classifieds with LLR and all fields query model (T+D+A+C-LLR) in the query model and BM25, number of examples retrieve uniquely this number of classifieds with all fields query model (T+D+A+C) and BM25.

Relevant classifieds	T+D+A+C-LLR	T+D+A+C
1	39	38
2	10	12
3	2	1

No big difference is obvious thus this subject will be added on the future work due to time **constrains**. One possible explanation that precision is harmed is that the the same relevant classifieds are scored less when LLR is used due to less word matching. Thus, they retrieve the same relevant classifieds but with different rank.

Query models with pseudo relevance feedback

Pseudo relevance feedback extends the query and more relevant classifieds retrieved. To prove this assumption, we present the following table with two different systems.

Table 6.9: System Performance (P@5) of all fields (T+D+A+C) and all fields using pseudo relevance feedback (T+D+A+C-Pseudo) query models and two types of ground truth (editorial and click logs).

	Editorial			Click logs		
	BM25	LM	TfIdf	BM25	LM	TfIdf
T+D+A+C	0.7300	0.6460	0.6860	0.1720	0.1700	0.1580
Pseudo(T+D+A+C)	0.7300	0.4920	0.6520	0.1720	0.1620	0.1500

The results show that in the case of LM and TfIdf there is a decrease in the P@5, but in the case of Okapi it remains stable. Since the pseudo relevance feedback is an expensive procedure due to the fact that it needs to retrieve results, take feedback and then to retrieve new result list, we decide that we will not use pseudo relevance feedback to the rest of the query models.

6.3 Retrieval methods

In the results provided in results chapter, we can see that Okapi BM25 performs better than the other two. In the following table we further prove the same assumption. We present the count of relevant and irrelevant results retrieved of each retrieval strategy.

Table 6.10: Retrieval strategies (BM25, LM, TfIdf), count of relevant results and count of irrelevant results retrieved from each one.

	Relevant	Irrelevant
BM25	3121	1879
LM	2399	2101
TfIdf	2633	1858

It is obvious that Okapi BM25 retrieves the biggest number of relevant results. Though, TfIdf retrieves 21 less irrelevant results. However, the fact that the precision of Okapi BM25 is always the greatest in all query models, makes us enough confident to say that Okapi BM25 is better than the other two.

6.4 Late Data Fusion experiments

In the table 5.3, we can see that late data fusion methods are not performing better than the best individual system. We were expecting that combMIN will have the worst precision since it takes into account the smallest score. But, we believed that combMNZ will perform better than the individual due to the fact that it boosts score and rank of a relevant document that is agreed upon many systems.

From the other side, using the click logs evaluation the precision is improved on the weighted fusion methods (WcombSUM, WcombMNZ, WcombWW), but the difference is just 0.02 which is not enough to make us confident to say that they perform better.

Our assumption is that we have these results because we fused all the systems together. Some of them they are not performing as well. To prove that assumption, we present the following table with the total number of relevant and irrelevant classifieds retrieved by a specific number of systems.

Table 6.11: Number of systems, total number of relevant and irrelevant classifieds retrieved by the specific number of systems.

Systems	Irrelevant	Relevant
1	10229	135
2	4034	115
3	1614	75
4	1168	49
5	704	57
6	549	71
7	419	61
8	312	74
9	266	47
10	227	53
11	154	37
12	143	46
13	109	59
14	97	41
15	79	46
16	65	36
17	45	21
18	47	31
19	38	38
20	31	34
21	32	41
22	38	34
23	19	42
24	19	33
25	4	32
26	1	20
27	1	25
28	1	31

The previous results indicate that there is only three irrelevant classified that is retrieved from more than 25 systems. Also, less count of systems retrieved a lot of irrelevant classifieds. From the other side, a lot of relevant results are retrieved only by one system (e.g. 135 relevant classifieds retrieved only by one system). Furthermore, only 31 relevant results retrieved by all all the systems. The different information in the query models is a possible reason of these

results but it will be part of future's work investigation.

To further analyze these results, we conduct one extra experiment with combANZ keeping stable all systemic differences like stemming, query modeling but retrieval strategy. We compare the results with combANZ result from the fusion of all individual systems to see if any improvement occurs.

Table 6.12: Systems used on the combANZ with stable systemic differences are presented. Explanation is given for the use of fields use in the query model, log likelihood ratio(LLR), stemming and pseudo relevance feedback (PRF).

Name	Fields	Stemming	LLR	PRF
T	title	no	no	no
T-LLR	title	no	yes	no
T stemming	title	yes	no	no
T-LLR stemming	title	yes	yes	no
T+D	title, description	no	no	no
T+D+A+C	title, description, category, attributes	no	no	no
A+C	attributes, category	no	no	no
T+D+A+C-LLR	title, description, category, attributes	no	yes	no
T+D+A+C-Pseudo	title, description, category, attributes	no	no	yes

Table 6.13: System Performance (P@5) of compANZ keeping stable systemic differences in a comparison with combANZ merged from all systems using editorial and click logs evaluation. Query models used are title (T), title using LLR (T-LLR), title using stemming (T-stemming), title and description (T+D), all the fields (T+D+A+C) and attribute and categories (A+C), all the fields using LLR (T+D+A+C-LLR), all the fields using pseudo relevance feedback (T+D+A+C-pseudo) query models and two types of ground truth (editorial and click logs).

Query model	Editorial	Click logs
T	0,2740	0.0700
T-LLR	0,2600	0.0520
T stemming	0,1520	0.0180
T-LLR stemming	0,0800	0.0060
T+D	0,2660	0.0220
T+D+A+C	0,2120	0.0200
A+C	0,1600	0.0080
T+D+A+C-LLR	0,1780	0.0280
T+D+A+C-Pseudo	0,1440	0.0280
combANZ	0.0800	0.0020

As you can see in the previous table, the combANZ is improved when the systemic differences kept stable. However they didn't outperform the precision of the individuals(see table 5.1).

To conclude, the fusion of the results **doesn't outperforms** the performance of the best individual system. However, we achieved to improve the effectiveness of combANZ when systemic differences kept stable. Thus, in the future plans, we can do the same experiment for the rest of the fusion method as well.

6.5 Diversification experiments



As you can see in the bar graphs in the results chapter, the differences in precision using either editorial or click logs evaluation are **too** small. Our assumption is that our results are too diversified already. This assumption can be proved if the precision in the individual systems is the same with the diversified systems. The following table presents the results of BM25 individual systems and BM25 diversified systems using editorial evaluation and click logs evaluation.

Table 6.14: System Performance (P@5) of diversified and individual systems using title (T), title and description (T+D), all the fields (T+D+A+C) and attribute and categories (A+C) query models, okapi BM25 retrieval strategy (BM25) and two types of ground truth (editorial and click logs).

	Editorial evaluation		Click logs evaluation	
	Individuals	Diversified	Individuals	Diversified
T	0.6560	0.6580	0.1680	0.1680
(T+D)	0.6660	0.6700	0.1700	0.1700
(T+D+A+C)	0.7300	0.7300	0.1720	0.1760
(A+C)	0.4800	0.4460	0.0920	0.0900

The differences in precision are really low in both evaluation approaches. We believe that the reason is that the initially retrieved results from the individual strategies are already **enough** diversified.



MMRalt1

The fact that the precision is not affected by this diversification approach is a positive sign. We take into account only the last one selected instead of all the previous. From the other side that can be a prove that our classifieds are enough diversified that are not affected by any diversification method.

MMRalt2

In this approach we used the average similarity in the calculation of the new score instead of choosing the maximum one. Thus, this approach can be used as an alternative since it doesn't affect the precision.

MMRaltAvgLast4

Same with the MMRalt1 approach, is a good sign that the precision is not affected by this algorithm due to the fact that we are comparing only the previous four selected results.

6.6 Data fusion of diversified result experiment

The results **on** the 5.4 table using editorial evaluation show us an improvement on the precision when the diversified lists are fused. Also, combMAX outperforms even our best individual system (T+D+A+D). Click logs evaluation from the other side shows a small difference in precision that doesn't indicate an actual improvement. However, combMAX's precision has a big boost of 0.1280 to 0.1700 which needs investigation to a future work. All in all, to answer the seventh research question, the fusion of diversified result lists improve the precision of the top five results, thus it performs better than the rest.

Chapter 7

Conclusion and Future work

presented

We **present** multiple experiments to find the best performing system for retrieving similar classifieds. The similar classifieds are based on the contents of the previous classified the user was checking.

The experiments **was** made in an industrial environment (a company) thus some of our decisions are affected by them. The company **provide** us with all the data we need to implement the experiments. They also **provide** us **of** assessors for the evaluation of the experiments.

We **present** multiple experiments to find the best performing system for retrieving similar classifieds. First, we **make** experiments to find the best query model in a comparison with the title query model which is the one the company is already using and our baseline. Then, we **investigate** which is the best retrieval strategy. Also, we **fuse** the available systems to improve the performance of individuals systems. Furthermore, we **make** experiments for diversification of the results. Finally, we **investigate** if the fusion of the diversified results **is improving** the results. We **evaluate** our experiments' results using two different indication of relevance, the editorial which was based on assessors opinion on relevance and click logs which was based on users history.

Results from the stemming experiment proved us that it harms the performance of the systems. Also, we present a big number of relevant results are not retrieved when stemming is used. Thus, we decide to **don't** use it in any other query model. In future work, we would like to investigate even further the reason that stemming decrease the precision. We believe that exposing queries before stemming and after and the results affected by them, will give a better insight.

Experimental evidence proves that **as much** information we are adding to the query model **as** better the system performs. Thus, the query model based on the entire classified content performs the best. Also, during the editorial evaluation that we examine visited classifieds and lists with similar classifieds, we observed that attributes and category adds a lot of information to the query. However, when **is** used alone (without the title and description terms) **is** not enough to cover the information need. Furthermore, the experimental results shows us also that attributes and category alone in the query model is the worst performed system. In future work, we can use the search query a user did to find the visited classified in the query model as well as extra information to cover his information need. Also, we will investigate all the possible combination of the fields on query modeling like adding the description in the attributes and category query model and see if any improvement occurs.

On an extra experiment **we make we prove** that choosing discriminative terms instead of the all the terms of the classified is not performing as good in terms of precision. Also, no big difference is obvious on the amount of relevant results retrieved by the same systems using



LLR and without. Thus, we would like to investigate further in the future the difference in the results of systems using or not the LLR approach. Also, we would like to do more experiments on discriminative terms using alternative methods.

The results of the pseudo relevance feedback experiment, show us no improvement in the precision. Thus, we didn't use it in more query models.

On retrieval method experiments, okapi BM25 performs better than the other two retrieval models in both type of evaluations and in all query models. Thus, it leaves us no doubt about which is the best retrieval method to improve our systems performance.

The experimental results for late data fusion experiments prove us that none of the late fusion methods is improving the precision of the best individual system. However, in an attempt to analyze it further and prove the reason that the results are not as expected, we conduct an extra experiment of combANZ keeping stable the systemic differences. With this experiment we improve the compANZ performance but it was not performing better than the best individual. Thus, we are planning to do the same experiment for the rest of the fusion methods in a future work.

Also, we compare four diversification approaches to find the best performing one. None of them show any difference in performance neither using editorial nor click logs evaluation. However, diversification is an expensive procedure due to the fact that you have to check the text similarity between all classifieds in the result list. Thus, the first alternative diversification method (MMRalt1) that compares only the previous selected classified with all the non selected, can be proved better because it doesn't affect the precision and is faster. Same is for the third alternative method (MMRaltAvgLast4) which takes into account only the previous four selected classifieds. But the assumption that these are faster and the fact that this makes them better needs a further investigation and a proper benchmark to be proved. Also, we already made the assumption that maybe the results are already diversified and that's why we don't have any big difference on the precision of the top five first results. In future work we will investigate the similarity of the classified to prove if this assumption is true or not.

Finally, we experiment with the fusion of the diversified results and we compare them initially with the fused individual systems. The results indicate that fusion of diversified results improves the precision. We also compare the results with the best individual system from the query model experiment and it's proved that combMAX improves even the performance of the best individual system. In future plans, we will compare these refinements with the diversification of fused results from [47].

To conclude, we achieved to improve the precision of the similar classifieds baseline's query model. We also find the best retrieval strategy that results the greatest precision. We proved that fusion of results is not performing better than our individual best system but we achieved to improve the precision of one of the fusion methods. We proposed three alternative diversification methods but none of them had big improvement in a comparison with the MMR of [35]. Finally, we fused the diversified results and we achieved to have the greatest precision of all the experiments.

Bibliography

- [1] Valentin Jijkoun , Gilad Mishne , Maarten de Rijke, *Preprocessing Documents to Answer Dutch Questions*, In proceedings of the 15th belgian-dutch conference on artificial intelligence BNAIC03, pp.487-497



[2] Parsing



[3] Ir book

[4] Stopwords

[5] Stemming

- [6] Hao Yang, Shuaib Ding, *Inverted Index Compression and Query Processing with Optimized Document Ordering*, In proceedings of the 18 International conference of World wide web, WWW '09, pp. 401-410

- [7] Shouning Qu, Sujuan Wang, Yann Zou, *Improvement of Text Feature Selection Method based on TFIDF*, In proceedings of the 2008 International Seminar on Future, Information Technology and Management Engineering, FITME '08, pp. 79-81

- [8] Fang Hui, Tao Tao, Zhai Chengxiang, *Diagnostic Evaluation of Information Retrieval Models*, ACM Transactions on Information Systems (TOIS) - Special issue on research and development in information retrieval TOIS Homepage archive, July 1991, pp 187 - 222

[9] Okapi

- [10] W. Bruce Croft, John Lafferty, *Language Modeling for Information Retrieval*, Springer Publishing Company, 2010

- [11] Chengxiang Zhai, John Lafferty, *A Study of Smoothing Methods for Language Models Applied to Ad Hoc Information Retrieval*, Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval, SIGIR '01, pp 334 - 342

[12] Relevance Feedback

- [13] Amit Singhal, *Modern Information Retrieval: a brief overview by Amit Singhal*, Bulletin of the IEEE computer society technical committee on data engineering

- [14] Paul Rayson, Roger Garside, *Comparing Corpora using Frequency Profiling*, In proceedings of the workshop on Comparing Corpora, held in conjunction ACL 2000, October 2000, Hong Kong, pp 1-6

- [15] Wilkins Peter, *An Investigation Into Weighted Data Fusion for Content-Based Multimedia Information Retrieval* PhD thesis, Dublin City University, Nov 2009
- [16] Data fusion
- [17] Christian Middleton, Ricardo Baeza-yates, *A Comparison of Open Source Search Engines*, SIGIR 2007
- [18] lemur
- [19] Trevor Strohman *Dynamic collections in Indri*
- [20] Stemmer
- [21] LLR
- [22] *Comparing Corpora using Frequency Profiling*
- [23] Chris Buckley, Darrin Dimmick, Ian Soboroff, Ellen Voorhees *Bias and the Limits of Pooling for Large Collections* Information Retrieval December 2007, pp 491-508
- [24] Stephen Robertson *On the history of evaluation in IR* Journal of Information Science August 2008, pp 439-456
- [25] Tetsuya Sakai, Teruko Mitamura *Boiling Down Information Retrieval Test Collections* RIAO '10 Adaptivity, Personalization and Fusion of Heterogeneous Information France 2010, pp 49-56
- [26] Craig Macdonald, Iadh Ounis, Ian Soboroff *Overview of the TREC-2009 Blog Track* In Proceedings of TREC 2009
- [27] Kalervo Järvelin, Jaana Kekkonen *IR evaluation methods for retrieving highly relevant documents*
- [28] Tefko Saracevic *Relevance: A Review of the Literature and a Framework for Thinking on the Notion in Information Science. Part III: Behavior and Effects of Relevance* Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2008, Singapore, July 20-24, 2008
- [29] Azzah Al-Maskari, Mark Sanderson, Paul Clough *Relevance Judgments between TREC and Non-TREC Assessors* November 2007, pp 2126 - 2144
- [30] Jean Carletta *Assessing Agreement on Classification Tasks: The Kappa Statistic* Computational Linguistics
- [31] Donna Harman *Overview of the First Text REtrieval Conference (TREC-1)* National Institute of Standards and Technology, Gaithersburg
- [32] Donna Harman *Overview of the Second Text REtrieval Conference (TREC-2)* National Institute of Standards and Technology Gaithersburg
- [33] James P Callan and W Bruce Croft *An Evaluation of Query Processing Strategies Using the TIPSTER Collection* In Proceedings of the 16th annual international ACM SIGIR Conference on Research and Development in Information Retrieval

- [34] H.P. Luhn *A statistical approach to mechanized encoding and searching of literary information* IBM Journal Research and Development, 1957
- [35] J. Carbonell and J. Goldstein *The use of MMR, diversity-based reranking for reordering documents and producing summaries* In Research and Development in Information Retrieval, 1998
- [36] R. Agrawal, S. Gollapudi, A. Halverson and S. Jeong *Diversifying search results* Proceedings of the 16th international conference on World Wide Web, WWW '07
- [37] C. Zhai and J. Lafferty *A Risk Minimization Framework for Information Retrieval* Information Processing and Management: an International Journal, 2006
- [38] H. Chen and D. Karger *Less is More-Probabilistic Models for Retrieving Fewer Relevant Documents* In Proceedings of the ACM Conference on Research and Development in Information Retrieval, 2006
- [39] M. Drosou and E. Pitoura *POIKILO: A Tool for Evaluating the Results of Diversification Models and Algorithms* Proceedings of the VLDB Endowment VLDB Endowment, Volume 6 Issue 12, August 2013
- [40] P. Chandar and B. Carterette *Analysis of Various Evaluation Measures for Diversity* Proceedings of the 1st International Workshop on Diversity in Document Retrieval at European Conference on Information Retrieval, ECIR11
- [41] N. J. Belkin , P. Kantor , E. A. Fox and J. A. Shaw *Combining the evidence of multiple query representations for information retrieval* Information Processing and Management, 1995
- [42] J. H. Lee *Analyses of Multiple Evidence Combination* Proceedings of the 20th annual international ACM SIGIR conference on Research and development in information retrieval, SIGIR '97
- [43] A. Chowdhury, O. Frieder, D. Grossman and C. McCabe *Analyses of Multiple-Evidence Combinations for Retrieval Strategies* Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval, SIGIR '01
- [44] S. M. Beitzel, E. C. Jensen, A. Chowdhury, O. Frieder, D. Grossman, N. Goharian *Disproving the Fusion Hypothesis: An Analysis of Data Fusion via Effective Information Retrieval Strategies* Proceedings of the 2003 ACM Symposium on Applied Computing
- [45] S. M. Beitzel, E. C. Jensen, A. Chowdhury, D. Grossman, O. Frieder, N. Goharian *On fusion of effective retrieval strategies in the same information retrieval system* Journal of the American Society of Information Science and Technology, 2004
- [46] R. Nuray and F. Can *Automatic ranking of information retrieval using data fusion* Information Processing and Management, 2006
- [47] S. Liang, Z. Ren and M. de Rijke *Fusion helps diversification* 37th international ACM SIGIR conference on Research and development in information retrieval, 2014

- [48] Abhinandan S. Das, Mayur Data, Ashutosh Garg and Shyam Rajaram *Google news personalization: scalable online collaborative filtering* WWW '07 Proceedings of the 16th international conference on World Wide Web, Pages 271-280
- [49] Nico Schlitter and Tanja Falkowski. *Mining the Dynamics of Music Preferences from a Social Networking Site* Social Network Analysis and Mining, 2009. ASONAM '09. International Conference on Advances, Pages: 243 - 248
- [50] Sean Owen, Robin Anil, Ted Dunning and Ellen Friedman *Mahood in action*
- [51] Eric Hatcher, Otis Cospodnetic and Michael McCandless *Lucene in action*
- [52] *Web Data Mining Research: A Survey* Computational Intelligence and Computing Research (ICCIC), 2010 IEEE International Conference on Pages: 1-10



Chapter 8

Appendix

8.1 Query modeling

8.1.1 Results Relevance ground truth

System	P5	P10	MAP	Rpr
T-TfIdf	0.626	0.484	0.2704	0.3023
T-LM	0.5980	0.4240	0.2480	0.2769
T-Okapi	0.6560	0.4910	0.2821	0.3115
T-TfIdf-LLR	0.5280	0.3840	0.2223	0.2474
T-LM-LLR	0.5160	0.3610	0.2162	0.2398
T-Okapi-LLR	0.5660	0.4020	0.2322	0.2543
T-TfIdf-stemming	0.5640	0.4280	0.2388	0.2705
T-LM-stemming	0.5660	0.4020	0.2320	0.2597
T-Okapi-stemming	0.6040	0.4410	0.2507	0.2788
T-TfIdf-stemming-LLR	0.4980	0.3600	0.2087	0.2336
T-LM-stemming-LLR	0.5160	0.3570	0.2121	0.2378
T-Okapi-stemming-LLR	0.5280	0.3860	0.2206	0.2491
(T+D) TfIdf	0.6500	0.4790	0.2716	0.2995
(T+D) LM	0.5880	0.4360	0.2460	0.2818
(T+D) Okapi	0.6660	0.4970	0.2809	0.3109
(T+D+A+C) TfIdf	0.6860	0.5280	0.2995	0.3347
(T+D+A+C) LM	0.6460	0.4600	0.2678	0.2966
(T+D+A+C) Okapi	0.7300	0.5190	0.3090	0.3357
(A+C) TfIdf	0.4500	0.2920	0.1463	0.1768
(A+C) LM	0.3400	0.2200	0.1148	0.1439
(A+C) Okapi	0.4800	0.2960	0.1534	0.1826
LLR(T+D+A+C) TfIdf	0.5740	0.3950	0.2213	0.2496
LLR(T+D+A+C) LM	0.5400	0.3720	0.2105	0.2360
LLR(T+D+A+C) Okapi	0.6360	0.4250	0.2402	0.2641
Pseudo(U+S) TfIdf	0.6520	0.4710	0.2731	0.3041
Pseudo(U+S) LM	0.4920	0.3000	0.1836	0.1995
Pseudo(U+S) Okapi	0.7300	0.5270	0.3170	0.3399
Pseudo(T) Okapi	0.6420	0.4600	0.2690	0.2992

8.1.2 Results Click ground truth

System	P5	P10	MAP	Rpr
T-TfIdf	0.1540	0.0920	0.5305	0.5135
T-LM	0.1660	0.0930	0.5610	0.5558
T-Okapi	0.1680	0.1020	0.5653	0.5563
T-TfIdf-LLR	0.1500	0.0790	0.5125	0.4909
T-LM-LLR	0.1540	0.0840	0.5314	0.5065
T-Okapi-LLR	0.1580	0.0850	0.5341	0.5166
T-TfIdf-stemming	0.1600	0.0900	0.5376	0.5273
T-LM-stemming	0.1660	0.0920	0.5582	0.5489
T-Okapi-stemming	0.1640	0.0940	0.5584	0.5492
T-TfIdf-stemming-LLR	0.1460	0.0790	0.4868	0.4604
T-LM-stemming-LLR	0.1560	0.0840	0.5295	0.5050
T-Okapi-stemming-LLR	0.1580	0.0860	0.5374	0.5162
(T+D) TfIdf	0.1600	0.0910	0.5321	0.5155
(T+D) LM	0.1680	0.0980	0.5629	0.5587
(T+D) Okapi	0.1700	0.1040	0.5619	0.5536
(T+D+A+C) TfIdf	0.1580	0.0950	0.5644	0.5606
(T+D+A+C) LM	0.1700	0.0960	0.5726	0.5692
(T+D+A+C) Okapi	0.1720	0.0990	0.5764	0.5675
(A+C) TfIdf	0.0600	0.0410	0.1810	0.1507
(A+C) LM	0.0900	0.0540	0.2946	0.2687
(A+C) Okapi	0.0920	0.0590	0.2987	0.2734
LLR(T+D+A+C) TfIdf	0.1540	0.0890	0.5220	0.5061
LLR(T+D+A+C) LM	0.1640	0.0880	0.5351	0.5053
LLR(T+D+A+C) Okapi	0.1600	0.0920	0.5389	0.5170
Pseudo(U+S)TfIdf	0.1500	0.0900	0.5298	0.5237
Pseudo(U+S)LM	0.1620	0.0860	0.5496	0.5459
Pseudo(U+S)Okapi	0.1720	0.0990	0.5741	0.5689
Pseudo(T)	0.1180	0.0750	0.3787	0.3454

8.2 LDF

8.2.1 LDF - Relevance Feedback

System	P5	P10	MAP	Rpr
combMAX	0.5400	0.3730	0.3426	0.2919
combMIN	0.0760	0.0460	0.0723	0.0359
combSUM	0.6160	0.4980	0.4589	0.4160
combMNZ	0.6460	0.5030	0.4688	0.4309
combANZ	0.0800	0.0490	0.0858	0.0374
WcombSUM	0.6140	0.4920	0.4511	0.4135
WcombMNZ	0.6360	0.5050	0.4663	0.4294
WcombWW	0.6320	0.5080	0.4657	0.4281

8.2.2 LDF - Clicks Ground Truth

System	P5	P10	MAP	Rpr
combMAX	0.1280	0.0730	0.4393	0.4093
combMIN	0.0020	0.0020	0.0080	0.0012
combSUM	0.1720	0.1030	0.5683	0.5515
combMNZ	0.1720	0.1040	0.5619	0.5455
combANZ	0.0020	0.0020	0.0119	0.0018
WcombSUM	0.1740	0.1040	0.5636	0.5438
WcombMNZ	0.1740	0.1030	0.5621	0.5445
WcombWW	0.1740	0.1040	0.5630	0.5456

8.2.3 combANZ - Click ground truth

query	queryno	P@5	P@10	MAP	Rpr
T	1	0.0700	0.0570	0.1535	0.1010
T-LLR	2	0.0520	0.0520	0.1223	0.0764
T stemming	3	0.0180	0.0110	0.0662	0.0342
T-LLR stemming	4	0.0060	0.0070	0.0293	0.0036
T+D	5	0.0220	0.0210	0.0644	0.0164
T+D+A+C	6	0.0200	0.0320	0.0718	0.0161
A+C	7	0.0080	0.0090	0.0282	0.0091
LLR(T+D+A+C)	8	0.0280	0.0340	0.0697	0.0160
Pseudo U+S	9	0.0280	0.0300	0.0911	0.0549

8.2.4 combANZ - Relevance ground truth

query	queryno	P@5	P@10	MAP	Rpr
T	1	0.2740	0.2600	0.2004	0.2561
T-LLR	2	0.2600	0.2440	0.1692	0.2289
T stemming	3	0.1520	0.1120	0.1242	0.1249
T-LLR stemming	4	0.0800	0.0620	0.0765	0.0680
T+D	5	0.2660	0.2090	0.1881	0.2160
T+D+A+C	6	0.2120	0.2190	0.1775	0.2368
A+C	7	0.1600	0.0990	0.0731	0.0849
LLR(T+D+A+C)	8	0.1780	0.1670	0.1264	0.1754
Pseudo U+S	9	0.1440	0.1400	0.1177	0.1122

8.3 LDF-MMR

8.3.1 LDF Results-MMR

LDF Results-MMR-Relevance ground truth

System	P5	P10	MAP	Rpr
combANZ	0.0940	0.0860	0.1078	0.0826
compMAX	0.7320	0.5290	0.4679	0.4237
combMIN	0.0800	0.0570	0.0662	0.0358
combMNZ	0.6700	0.5290	0.4670	0.4400
combSUM	0.6700	0.5270	0.4669	0.4400
WcombMNZ	0.6560	0.5320	0.4649	0.4389
WcombSUM	0.6620	0.5400	0.4675	0.4399
WcompWW	0.6560	0.5370	0.4658	0.4412

8.3.2 LDF Results-MMRAIt1

LDF Results-MMRAIt1-Relevance ground truth

System	P5	P10	MAP	Rpr
combANZ	0.1320	0.1050	0.1184	0.0978
compMAX	0.7280	0.5380	0.4748	0.4279
combMIN	0.0820	0.0540	0.0662	0.0341
combMNZ	0.6760	0.5280	0.4762	0.4390
combSUM	0.6760	0.5280	0.4762	0.4390
WcombMNZ	0.6720	0.5250	0.4782	0.4415
WcombSUM	0.6740	0.5270	0.4815	0.4460
WcompWW	0.6740	0.5280	0.4814	0.4465

8.3.3 LDF Results-MMRAIt2

LDF Results-MMRAIt2-Relevance ground truth

System	P5	P10	MAP	Rpr
combANZ	0.1800	0.1300	0.1417	0.1220
compMAX	0.7140	0.5390	0.4671	0.4256
combMIN	0.0800	0.0520	0.0660	0.0347
combMNZ	0.6700	0.5290	0.4753	0.4396
combSUM	0.6700	0.5290	0.4754	0.4396
WcombMNZ	0.6600	0.5240	0.4713	0.4411
WcombSUM	0.6640	0.5280	0.4739	0.4450
WcompWW	0.6680	0.5310	0.4794	0.4491

8.3.4 LDF Results-MMRAItAvgLast4

LDF Results-MMRAItAvgLast4-Relevance ground truth

System	P5	P10	MAP	Rpr
combANZ	0.1760	0.1260	0.1374	0.1213
compMAX	0.7120	0.5410	0.4665	0.4280
combMIN	0.0800	0.0520	0.0656	0.0341
combMNZ	0.6720	0.5280	0.4753	0.4391
combSUM	0.6720	0.5280	0.4754	0.4391
WcombMNZ	0.6620	0.5230	0.4721	0.4420
WcombSUM	0.6620	0.5250	0.4749	0.4434
WcompWW	0.6640	0.5330	0.4811	0.4486

8.4 Diversification

8.4.1 MMR

Results Relevance ground truth-MMR

T-TfIdf-MMR	0.6080	0.4760	0.3267	0.3709
T-LM-MMR	0.4520	0.3170	0.2425	0.2853
T-Okapi-MMR	0.6580	0.4920	0.3404	0.3778
T-TfIdf-LLR-MMR	0.4880	0.3610	0.2457	0.2953
T-LM-LLR-MMR	0.4300	0.3050	0.2134	0.2629
T-Okapi-LLR-MMR	0.5600	0.3990	0.2717	0.3118
T-TfIdf-stemming-MMR	0.5500	0.4250	0.2884	0.3352
T-LM-stemming-MMR	0.4240	0.2910	0.2224	0.2752
T-Okapi-stemming-MMR	0.6100	0.4400	0.3080	0.3484
T-TfIdf-stemming-LLR-MMR	0.4620	0.3300	0.2365	0.2883
T-LM-stemming-LLR-MMR	0.4340	0.2960	0.2091	0.2538
T-Okapi-stemming-LLR-MMR	0.5240	0.3910	0.2621	0.3047
(T+D) TfIdf-MMR	0.6080	0.4470	0.3159	0.3590
(T+D) LM-MMR	0.4540	0.3300	0.2413	0.2889
(T+D) Okapi MMR	0.6700	0.4930	0.3421	0.3801
(T+D+A+C) TfIdf MMR	0.6660	0.5060	0.3501	0.3933
(T+D+A+C) LM MMR	0.4960	0.3520	0.2605	0.3103
(T+D+A+C) Okapi MMR	0.7300	0.5200	0.3596	0.3957
(A+C) TfIdf MMR	0.3440	0.2350	0.1427	0.1886
(A+C) LM-MMR	0.2460	0.1820	0.1009	0.1470
(A+C) Okapi-MMR	0.4460	0.2870	0.1625	0.2017
LLR(T+D+A+C) TfIdf-MMR	0.4940	0.3640	0.2418	0.2911
LLR(T+D+A+C) LM-MMR	0.4540	0.3160	0.2160	0.2697
LLR(T+D+A+C) Okapi-MMR	0.5820	0.4130	0.2699	0.3087
Pseudo(U+S) TfIdf-MMR	0.6460	0.4710	0.3285	0.3700
Pseudo(U+S) LM-MMR	0.2380	0.1820	0.1284	0.1849
Pseudo(U+S) Okapi-MMR	0.7400	0.5340	0.3835	0.4160
Pseudo(T)-MMR	0.6800	0.4840	0.3239	0.3604

Results Click ground truth-MMR

System	P5	P10	MAP	Rpr
T-TfIdf-MMR	0.1580	0.0930	0.5341	0.5213
T-LM-MMR	0.1540	0.0860	0.5251	0.4981
T-Okapi-MMR	0.1680	0.1020	0.5650	0.5563
T-TfIdf-LLR-MMR	0.1460	0.0800	0.5101	0.5014
T-LM-LLR-MMR	0.1440	0.0790	0.5001	0.4785
T-Okapi-LLR-MMR	0.1580	0.0840	0.5377	0.5277
T-TfIdf-stemming-MMR	0.1580	0.0890	0.5362	0.5268
T-LM-stemming-MMR	0.1460	0.0800	0.5210	0.5019
T-Okapi-stemming-MMR	0.1660	0.0940	0.5587	0.5487
T-TfIdf-stemming-LLR-MMR	0.1380	0.0780	0.4657	0.4382
T-LM-stemming-LLR-MMR	0.1460	0.0810	0.5103	0.4998
T-Okapi-stemming-LLR-MMR	0.1620	0.0860	0.5397	0.5329
(T+D) TfIdf-MMR	0.1580	0.0890	0.5324	0.5189
(T+D) LM-MMR	0.1440	0.0820	0.5309	0.5165
(T+D) Okapi MMR	0.1700	0.1060	0.5633	0.5547
(T+D+A+C) TfIdf MMR	0.1580	0.0950	0.5555	0.5451
(T+D+A+C) LM MMR	0.1560	0.0900	0.5539	0.5411
(T+D+A+C) Okapi MMR	0.1760	0.1000	0.5772	0.5697
(A+C) TfIdf MMR	0.0480	0.0360	0.1448	0.1102
(A+C) LM-MMR	0.0640	0.0410	0.2181	0.1835
(A+C) Okapi-MMR	0.0900	0.0580	0.2957	0.2634
LLR(T+D+A+C) TfIdf-MMR	0.1420	0.0840	0.4911	0.4693
LLR(T+D+A+C) LM-MMR	0.1480	0.0850	0.5108	0.4904
LLR(T+D+A+C) Okapi-MMR	0.1560	0.0920	0.5410	0.5270
Pseudo(U+S) TfIdf-MMR	0.1500	0.0890	0.5239	0.5121
Pseudo(U+S) LM-MMR	0.1060	0.0680	0.3990	0.3501
Pseudo(U+S) Okapi-MMR	0.1720	0.0990	0.5742	0.5689
Pseudo(T)-MMR	0.1180	0.0740	0.3780	0.3448

8.4.2 MMRAlt1

Results Relevance ground truth-MMRAlt

System	P5	P10	MAP	Rpr
T-TfIdf-MMRAlt	0.5120	0.4150	0.2593	0.3162
T-LM-MMRAlt	0.4160	0.3120	0.2036	0.2491
T-Okapi-MMRAlt	0.6540	0.4890	0.3458	0.3846
T-TfIdf-LLR-MMRAlt	0.4460	0.3350	0.2050	0.2599
T-LM-LLR-MMRAlt	0.3600	0.2700	0.1710	0.2226
T-Okapi-LLR-MMRAlt	0.5680	0.3980	0.2766	0.3175
T-TfIdf-stemming-MMRAlt	0.4080	0.2970	0.1778	0.2192
T-LM-stemming-MMRAlt	0.3740	0.2790	0.1813	0.2341
T-Okapi-stemming-MMRAlt	0.6020	0.4400	0.3155	0.3518
T-TfIdf-stemming-LLR-MMRAlt	0.2160	0.1600	0.0929	0.1257
T-LM-stemming-LLR-MMRAlt	0.3520	0.2630	0.1616	0.2068
T-Okapi-stemming-LLR-MMRAlt	0.5260	0.3870	0.2681	0.3121
(T+D) TfIdf-MMRAlt	0.5780	0.4440	0.2893	0.3384
(T+D) LM-MMRAlt	0.4420	0.3320	0.2080	0.2594
(T+D) Okapi MMRAlt	0.6660	0.4950	0.3488	0.3844
(T+D+A+C) TfIdf MMRAlt	0.5800	0.4430	0.2883	0.3363
(T+D+A+C) LM MMRAlt	0.4580	0.3640	0.2254	0.2855
(T+D+A+C) Okapi MMRAlt	0.7320	0.5180	0.3658	0.3994
(A+C) TfIdf MMRAlt	0.3480	0.2350	0.1283	0.1652
(A+C) LM-MMRAlt	0.2080	0.1540	0.0738	0.1127
(A+C) Okapi-MMRAlt	0.4600	0.2930	0.1688	0.2088
LLR(T+D+A+C) TfIdf-MMRAlt	0.4280	0.3150	0.1938	0.2455
LLR(T+D+A+C) LM-MMRAlt	0.3720	0.2700	0.1597	0.2113
LLR(T+D+A+C) Okapi-MMRAlt	0.6180	0.4200	0.2770	0.3138
Pseudo(U+S) TfIdf-MMRAlt	0.5420	0.4090	0.2681	0.3100
Pseudo(U+S) LM-MMRAlt	0.2660	0.1970	0.1135	0.1503
Pseudo(U+S) Okapi-MMRAlt	0.7320	0.5240	0.3847	0.4153
Pseudo(T)-MMRAlt	0.6400	0.4600	0.3128	0.3534

Results Click ground truth-MMRAIt

System	P5	P10	MAP	Rpr
T-TfIdf-MMRAIt	0.0480	0.0360	0.0704	0.0388
T-LM-MMRAIt	0.0440	0.0280	0.0561	0.0319
T-Okapi-MMRAIt	0.1680	0.1010	0.5653	0.5563
T-TfIdf-LLR-MMRAIt	0.0480	0.0310	0.0899	0.0621
T-LM-LLR-MMRAIt	0.0380	0.0280	0.0786	0.0563
T-Okapi-LLR-MMRAIt	0.1580	0.0850	0.5351	0.5166
T-TfIdf-stemming-MMRAIt	0.0400	0.0250	0.0649	0.0349
T-LM-stemming-MMRAIt	0.0420	0.0280	0.0632	0.0418
T-Okapi-stemming-MMRAIt	0.1660	0.0950	0.5588	0.5492
T-TfIdf-stemming-LLR-MMRAIt	0.0240	0.0180	0.0502	0.0399
T-LM-stemming-LLR-MMRAIt	0.0400	0.0310	0.0927	0.0726
T-Okapi-stemming-LLR-MMRAIt	0.1580	0.0850	0.5383	0.5178
(T+D) TfIdf-MMRAIt	0.0420	0.0330	0.0531	0.0372
(T+D) LM-MMRAIt	0.0560	0.0340	0.0581	0.0341
(T+D) Okapi MMRAIt	0.1720	0.1050	0.5627	0.5553
(T+D+A+C) TfIdf MMRAIt	0.0420	0.0340	0.0540	0.0372
(T+D+A+C) LM MMRAIt	0.0380	0.0360	0.0513	0.0354
(T+D+A+C) Okapi MMRAIt	0.1720	0.0990	0.5765	0.5687
(A+C) TfIdf MMRAIt	0.0220	0.0250	0.0565	0.0183
(A+C) LM-MMRAIt	0.0280	0.0220	0.0626	0.0344
(A+C) Okapi-MMRAIt	0.0900	0.0590	0.3002	0.2734
LLR(T+D+A+C) TfIdf-MMRAIt	0.0380	0.0310	0.0699	0.0437
LLR(T+D+A+C) LM-MMRAIt	0.0500	0.0330	0.0812	0.0552
LLR(T+D+A+C) Okapi-MMRAIt	0.1600	0.0920	0.5376	0.5165
Pseudo(U+S)TfIdf-MMRAIt	0.0360	0.0350	0.0521	0.0341
Pseudo(U+S) LM-MMRAIt	0.0340	0.0220	0.0405	0.0280
Pseudo(U+S)Okapi-MMRAIt	0.1720	0.0990	0.5741	0.5689
Pseudo(T)-MMRAIt	0.1180	0.0750	0.3788	0.3454

8.4.3 MMRAlt2

Results Relevance ground truth-MMRAlt2

System	P5	P10	MAP	Rpr
T-TfIdf-MMRAlt2	0.5020	0.4000	0.2571	0.3175
T-LM-MMRAlt2	0.3420	0.2700	0.1783	0.2332
T-Okapi-MMRAlt2	0.6560	0.4860	0.3424	0.3798
T-TfIdf-LLR-MMRAlt2	0.4140	0.3110	0.1895	0.2502
T-LM-LLR-MMRAlt2	0.2940	0.2130	0.1448	0.1960
T-Okapi-LLR-MMRAlt2	0.5500	0.3790	0.2693	0.3096
T-TfIdf-stemming-MMRAlt2	0.4120	0.3060	0.1796	0.2269
T-LM-stemming-MMRAlt2	0.2760	0.2180	0.1512	0.2078
T-Okapi-stemming-MMRAlt2	0.6040	0.4330	0.3130	0.3533
T-TfIdf-stemming-LLR-MMRAlt2	0.1980	0.1430	0.0822	0.1180
T-LM-stemming-LLR-MMRAlt2	0.2620	0.2020	0.1376	0.1840
T-Okapi-stemming-LLR-MMRAlt2	0.5200	0.3730	0.2624	0.3033
(T+D) TfIdf-MMRAlt2	0.5580	0.4140	0.2785	0.3217
(T+D) LM-MMRAlt2	0.3360	0.2520	0.1757	0.2254
(T+D) Okapi MMRAlt2	0.6580	0.4820	0.3426	0.3836
(T+D+A+C) TfIdf MMRAlt2	0.5580	0.4140	0.2785	0.3217
(T+D+A+C) LM MMRAlt2	0.3840	0.2940	0.1943	0.2535
(T+D+A+C) Okapi MMRAlt2	0.7240	0.5170	0.3640	0.3971
(A+C) TfIdf MMRAlt2	0.3060	0.2110	0.1132	0.1599
(A+C) LM-MMRAlt2	0.2300	0.1560	0.0775	0.1153
(A+C) Okapi-MMRAlt2	0.4480	0.2920	0.1674	0.2084
LLR(T+D+A+C) TfIdf-MMRAlt2	0.4020	0.3020	0.1858	0.2390
LLR(T+D+A+C) LM-MMRAlt2	0.3380	0.2520	0.1486	0.2147
LLR(T+D+A+C) Okapi-MMRAlt2	0.6080	0.4130	0.2746	0.3102
Pseudo(U+S) TfIdf-MMRAlt2	0.5300	0.3880	0.2586	0.2980
Pseudo(U+S) LM-MMRAlt2	0.1780	0.1430	0.0827	0.1364
Pseudo(U+S) Okapi-MMRAlt2	0.7180	0.5200	0.3765	0.4086
Pseudo(T)-MMRAlt2	0.6440	0.4640	0.3135	0.3531

Results Click ground truth-MMRAIt2

T-TfIdf-MMRAIt2	0.0580	0.0420	0.1063	0.0837
T-LM-MMRAIt2	0.0280	0.0250	0.0573	0.0343
T-Okapi-MMRAIt2	0.1680	0.0990	0.5647	0.5557
T-TfIdf-LLR-MMRAIt2	0.0380	0.0290	0.0835	0.0631
T-LM-LLR-MMRAIt2	0.0340	0.0220	0.0782	0.0540
T-Okapi-LLR-MMRAIt2	0.1540	0.0850	0.5343	0.5176
T-TfIdf-stemming-MMRAIt2	0.0640	0.0370	0.1513	0.1222
T-LM-stemming-MMRAIt2	0.0280	0.0180	0.0614	0.0284
T-Okapi-stemming-MMRAIt2	0.1660	0.0950	0.5582	0.5481
T-TfIdf-stemming-LLR-MMRAIt2	0.0200	0.0160	0.0532	0.0406
T-LM-stemming-LLR-MMRAIt2	0.0360	0.0230	0.0927	0.0654
T-Okapi-stemming-LLR-MMRAIt2	0.1600	0.0850	0.5361	0.5218
(T+D) TfIdf-MMRAIt2	0.0400	0.0340	0.0510	0.0355
(T+D) LM-MMRAIt2	0.0320	0.0190	0.0607	0.0298
(T+D) Okapi MMRAIt2	0.1700	0.1010	0.5605	0.5519
(T+D+A+C) TfIdf MMRAIt2	0.0400	0.0340	0.0510	0.0355
(T+D+A+C) LM MMRAIt2	0.0440	0.0270	0.0641	0.0390
(T+D+A+C) Okapi MMRAIt2	0.1720	0.0990	0.5773	0.5693
(A+C) TfIdf MMRAIt2	0.0280	0.0270	0.0576	0.0217
(A+C) LM-MMRAIt2	0.0280	0.0220	0.0695	0.0200
(A+C) Okapi-MMRAIt2	0.0920	0.0590	0.2977	0.2734
LLR(T+D+A+C) TfIdf-MMRAIt2	0.0400	0.0280	0.0800	0.0616
LLR(T+D+A+C) LM-MMRAIt2	0.0420	0.0330	0.0936	0.0636
LLR(T+D+A+C) Okapi-MMRAIt2	0.1620	0.0900	0.5385	0.5166
Pseudo(U+S) TfIdf-MMRAIt2	0.0360	0.0320	0.0503	0.0324
Pseudo(U+S) LM-MMRAIt2	0.0160	0.0130	0.0416	0.0180
Pseudo(U+S) Okapi-MMRAIt2	0.1560	0.0930	0.4806	0.4506
Pseudo(T)-MMRAIt2	0.1200	0.0740	0.3788	0.3454

8.4.4 MMRAltAvgLst4

Results Relevance ground truth-MMRAltAvgLast4

System	P5	P10	MAP	Rpr
T-TfIdf-MMRAltAvgLast4	0.5140	0.4060	0.2616	0.3198
T-LM-MMRAltAvgLast4	0.3360	0.2490	0.1706	0.2233
T-Okapi-MMRAltAvgLast4	0.6460	0.4900	0.3435	0.3825
T-TfIdf-LLR-MMRAltAvgLast4	0.4060	0.3230	0.1917	0.2535
T-LM-LLR-MMRAltAvgLast4	0.2820	0.2130	0.1434	0.1911
T-Okapi-LLR-MMRAltAvgLast4	0.5460	0.3820	0.2709	0.3140
T-TfIdf-stemming-MMRAltAvgLast4	0.3980	0.3030	0.1792	0.2260
T-LM-stemming-MMRAltAvgLast4	0.2900	0.2290	0.1531	0.2086
T-Okapi-stemming-MMRAltAvgLast4	0.6000	0.4420	0.3137	0.3505
T-TfIdf-stemming-LLR-MMRAltAvgLast4	0.2020	0.1440	0.0867	0.1251
T-LM-stemming-LLR-MMRAltAvgLast4	0.2740	0.2110	0.1390	0.1928
T-Okapi-stemming-LLR-MMRAltAvgLast4	0.5080	0.3730	0.2615	0.3066
(T+D) TfIdf-MMRAltAvgLast4	0.5560	0.4260	0.2842	0.3338
(T+D) LM-MMRAltAvgLast4	0.3420	0.2530	0.1731	0.2231
(T+D) Okapi MMRAltAvgLast4	0.6540	0.4810	0.3422	0.3809
(T+D+A+C) TfIdf MMRAltAvgLast4	0.5560	0.4260	0.2842	0.3338
(T+D+A+C) LM MMRAltAvgLast4	0.3700	0.2980	0.1915	0.2494
(T+D+A+C) Okapi MMRAltAvgLast4	0.7200	0.5200	0.3640	0.4028
(A+C) TfIdf MMRAltAvgLast4	0.2980	0.2050	0.1137	0.1517
(A+C) LM-MMRAltAvgLast4	0.1820	0.1380	0.0721	0.1092
(A+C) Okapi-MMRAltAvgLast4	0.4260	0.2930	0.1673	0.2089
LLR(T+D+A+C) TfIdf-MMRAltAvgLast4	0.4080	0.3030	0.1873	0.2377
LLR(T+D+A+C) LM-MMRAltAvgLast4	0.3240	0.2360	0.1446	0.1989
LLR(T+D+A+C) Okapi-MMRAltAvgLast4	0.5760	0.4060	0.2722	0.3108
Pseudo(U+S)TfIdf-MMRAltAvgLast4	0.5060	0.3860	0.2591	0.3052
Pseudo(U+S) LM-MMRAltAvgLast4	0.1460	0.1300	0.0786	0.1285
Pseudo(U+S) Okapi-MMRAltAvgLast4	0.7140	0.5220	0.3760	0.4098
Pseudo(T)-MMRAltAvgLast4	0.6380	0.4650	0.3126	0.3530

Results Click ground truth-MMRAltAvgLast4

T-TfIdf-MMRAltAvgLast4	0.0560	0.0410	0.0982	0.0743
T-LM-MMRAltAvgLast4	0.0380	0.0240	0.0509	0.0347
T-Okapi-MMRAltAvgLast4	0.1660	0.1000	0.5645	0.5547
T-TfIdf-LLR-MMRAltAvgLast4	0.0380	0.0300	0.0854	0.0621
T-LM-LLR-MMRAltAvgLast4	0.0320	0.0200	0.0724	0.0517
T-Okapi-LLR-MMRAltAvgLast4	0.1580	0.0860	0.5353	0.5166
T-TfIdf-stemming-MMRAltAvgLast4	0.0600	0.0370	0.1503	0.1232
T-LM-stemming-MMRAltAvgLast4	0.0320	0.0200	0.0529	0.0311
T-Okapi-stemming-MMRAltAvgLast4	0.1660	0.0930	0.5587	0.5481
T-TfIdf-stemming-LLR-MMRAltAvgLast4	0.0240	0.0140	0.0541	0.0433
T-LM-stemming-LLR-MMRAltAvgLast4	0.0360	0.0240	0.0890	0.0634
T-Okapi-stemming-LLR-MMRAltAvgLast4	0.1540	0.0850	0.5353	0.5201
(T+D) TfIdf-MMRAltAvgLast4	0.0420	0.0330	0.0514	0.0377
(T+D) LM-MMRAltAvgLast4	0.0380	0.0220	0.0522	0.0323
(T+D) Okapi MMRAltAvgLast4	0.1700	0.1020	0.5616	0.5519
(T+D+A+C) TfIdf MMRAltAvgLast4	0.0420	0.0330	0.0514	0.0377
(T+D+A+C) LM MMRAltAvgLast4	0.0340	0.0280	0.0498	0.0308
(T+D+A+C) Okapi MMRAltAvgLast4	0.1740	0.0990	0.5770	0.5692
(A+C) TfIdf MMRAltAvgLast4	0.0260	0.0200	0.0509	0.0190
(A+C) LM-MMRAltAvgLast4	0.0220	0.0210	0.0566	0.0215
(A+C) Okapi-MMRAltAvgLast4	0.0920	0.0590	0.2983	0.2734
LLR(T+D+A+C) TfIdf-MMRAltAvgLast4	0.0400	0.0290	0.0801	0.0614
LLR(T+D+A+C) LM-MMRAltAvgLast4	0.0400	0.0290	0.0844	0.0602
LLR(T+D+A+C) Okapi-MMRAltAvgLast4	0.1600	0.0910	0.5373	0.5147
Pseudo(U+S) TfIdf-MMRAltAvgLast4	0.0340	0.0320	0.0511	0.0347
Pseudo(U+S) LM-MMRAltAvgLast4	0.0160	0.0120	0.0348	0.0164
Pseudo(U+S) Okapi-MMRAltAvgLast4	0.1560	0.0930	0.4804	0.4506
Pseudo(T)-MMRAltAvgLast4	0.1180	0.0740	0.3782	0.3454