Information Retrieval Evaluation Measures Based on Preference Graphs

by

Chengxi Luo

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Mathematics
in
Computer Science

Waterloo, Ontario, Canada, 2021

Examining Committee Membership

The following served on the Examining Committee for this thesis. The decision of the Examining Committee is by majority vote.

Supervisor(s):	Charles Clarke Professor, David R. Cheriton School of Computer Science University of Waterloo
Internal Member:	Gordon Cormack Professor, David R. Cheriton School of Computer Science University of Waterloo
Internal Member:	Mark Smucker Associate Professor, Department of Management Sciences University of Waterloo

Author's Declaration

This thesis consists of material all of which I authored or co-authored: see Statement of Contributions included in the thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Statement of Contributions

This thesis includes the material which will be presented at SIGIR 2021 [13]. In particular, Chapter 4 includes text and analysis by my supervisor Prof. Charles Clarke for publication in that paper. I am the sole author of the remaining chapters.

Abstract

Offline evaluation for web search has used mostly graded judgments to evaluate the performance of information retrieval systems. While graded judgments suffer several known problems, preference judgments simply judge one item over another, which avoids the problem of complex definition of relevance scores. Previous research about evaluation measures for preference judgments focuses on translating preferences into relevance scores applied in the traditional evaluation measures, or weighting and counting the number of agreements between actual ranking from users' preferences and ideal ranking generated by systems. However, these measures lack clear theoretical foundations and their values have no obvious interpretation. On the other hand, although preference judgments for general web search have been studied extensively, there is limited research on investigating preference judgments application for web image search.

This thesis addresses exactly these questions, which proposes a preference-based evaluation measure to compute the maximum similarity between an actual ranking from users' preferences and an ideal ranking generated by systems. Specifically, this measure constructs a directed multigraph and computes the ordering of vertices, which we call the ideal ranking, that has maximum similarity to actual ranking calculated by the rank similarity measure. This measure is able to take any arbitrary collection of preferences that might include the property of conflicts, redundancies, incompleteness, and diverse type results (documents or images). Our results show that Greedy PGC is matches or exceeds the performance of evaluation measures proposed in previous research.

Acknowledgements

First and foremost, I would like to take this opportunity to express my sincere gratitude to my supervisor, Prof. Charles L.A. Clarke. Without his constant feedback and guidance, this research would not have been possible. He has been a great support for me throughout the last two years. Over the course of my studies, Prof. Clarke has taught me almost everything about research in the field of computer science. He was always available for technical discussions, helping me find my interested topic, and giving valuable suggestions to improve my writing and presentation skills. He directed me into the right way when I was lost. He is not only an academic advisor, but also a mentor in my life. I will treasure and remember all that I have learned from him.

I would like to thank members in the Data System Group lab. Group discussions motivate and help greatly my work. I would also like to thank my dear friend Songyun Cao for making me feel at home even though I was far away from home and giving continuous emotional support and encouragement when I was down.

Finally, I wish to acknowledge the support and great love of my family. Without their love and support, chasing my dreams and accomplishing everything that I have today would not have been possible.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

Offline evaluation for web search mainly uses graded judgments to represent the quality of search results and evaluate the performance of systems. Assessors are employed to give relevance scores independently for graded judgments following a relevance criteria. Evaluation measures for graded judgments translate these relevance scores into the effectiveness of web search engines. However, they have several known limitations even though graded judgements have been researched for a long history: 1) It is difficult to define clear and consistent grades of relevance [9]. 2) Assessors have to take huge effort to precisely decide the multi-level relevance grades. The complexity of labeling increases with the number of grades level and the number of other factors. 3) Existing work shows that there is no clear relationship between relevance scores and the level of user satisfaction so that higher relevance scores do not represent better user experience [4].

Preference judgments have been alternative to graded judgments. Instead of defining relevance grades and assigning relevance labels to each item in the result pool, preference judgements are that assessors examine two items, and prefer one item over another item. Preference judgments do not need to explicitly decide the number of grades and definitions of each grade. Carterette et al. [9] indicate that making preferences over two items is much easier than selecting one relevance score from a pre-defined relevance scale to one item for assessors to complete judgments. Preference judgments are less time consuming and they obtain better quality of judgments and agreement from the perspective of click rate and user satisfaction.

Although there have been considerable efforts on researching preference judgments and preference-based measures, there is no widely accepted and used evaluation method for preference judgments. Extending Carterette's work [9], Sakai and Zeng [40] define a va-

riety of preference-based measures and explore the relationship between these measures and users' preferences. These measures are based on counting a weighted agreement and disagreement of items in actual ranking generated by systems and a set of preferences, and translating these preferences into relevance scores applied in the traditional evaluation measures for graded judgments. However, giving weights to the number of agreements and translating preferences into relevance scores lack any clear theoretical foundation. It is not clear that these weights and the relevance scores translated from preferences are correlated with users' satisfaction and preferences. Their proposed methods cannot be proved to perform better than an average assessor.

In terms of providing the ranking of all items, preference judgments take more effort than graded judgments. Clarke et al. [15, 16] solve this problem by using partial preferences, identifying top-k rankings, and evaluating a system's performance by measuring maximum similarity between these rankings. They run an elimination tournament to generate a top-k actual ranking, give a detailed definition of ideal ranking, and choose Rank Biased Overlap (RBO) as the rank similarity measure to compare rankings. However, by only generating a top-k ranking, partial preferences discard the rest of preferences, which might miss some important information.

While most research focuses on preference judgments and preference-based evaluation measures for general web search, there is little work about preference judgments and preference-based evaluation measures for web image search. Unlike general web search that displays the result as a sequential list, web image search places the images in a grid style. To the best of our knowledge, Xie et al. [49] is the first attempt to study image search preference-based evaluation. They take grid-based assumptions [50] into consideration and propose a family of preference-based evaluation methods for web image search.

In this thesis, we combine the ideas of Sakai and Zeng, of Clarke et al., and of Xie et al. to propose a preference-based evaluation method, Greedy Preference Graph Compatibility (Greedy PGC), that works on an arbitrary collection of preferences. This collection allows diverse types of results (documents or images), conflicting items (conflict preferences coexist), redundant items (same preferences exist multiple times), incomplete items (rankings are mutually non-conjoint). This method takes a collection of preferences as a directed multi-graph, where each item is a vertex and each preference between items is an edge. It computes the ordering of vertices, which we call ideal ranking, that has maximum similarity to actual ranking calculated by the rank similarity measure RBO. For web image results, we incorporate grid-based assumptions with Greedy PGC to compute maximum similarity between a set of ideal rankings and an actual ranking. We apply Greedy PGC on datasets from Sakai and Zeng [40], from Clarke et al. [15], and from Xie et al. [49] and our results show that Greedy PGC matching or exceeding the performance of evaluation

measures proposed in these papers.

In the remainder of this thesis, we first introduce the offline evaluation measures for graded judgments and preference judgments in Chapter 2. In Chapter 3, we present an algorithm called Greedy PGC for system evaluation from preference judgments. It is developed based on the algorithm called Greedy Feedback Arc Set that outputs a vertex sequence with minimal backward arcs. In Chapter 4, we discuss the result of Greedy PGC on the datasets from Sakai and Zeng [40] and from Clarke et al. [15], and compare our results with their results. In Chapter 5, we introduce related work about preference judgments for web image search, and present our algorithm Greedy PGC for images. We also discuss the result of Greedy PGC on the image dataset from Xie et al. [49] and compare our result with their result. In Chapter 6, we summarize the contents and results of this thesis and provide directions for future work.

Chapter 2

Background

Online evaluation and offline evaluation are two of the most common measurements of evaluating the effectiveness of information retrieval systems. Katja et al. define online evaluation as "the evaluation of a fully functioning system based on implicit measurements of real users' experiences of the system in a natural usage environment" [27]. The basic core of implicit measurements is observing users' natural or normal interaction with the system [31]. Examples of implicit measurements are clicks, dwell times, and swipes. While these measurements are difficult and costly to control the variables and interpret the data [1, 33], offline evaluation takes less time to provide the outcome that is more easily to interpret. Unlike online evaluation that analyzes actual user behaviours and obtains indirect feedback, offline evaluation focuses on historical data or labeled data, and it is a direct comparison of systems.

2.1 Graded Judgements

Most offline evaluation metrics are based on relevance scores that represent the quality of search results from relevance judgement assessors. Relevance scores can be classified as binary relevance and multi-level relevance scales. Binary relevance evaluates whether an item is either relevant or not relevant to a query. If an item is relevant, it will have the same degree with other relevant items. Examples of binary relevance metrics include precision@k, mean reciprocal rank(MRR) and mean average precision(MAP). MAP was a primary evaluation methodology in the multiple tasks of early Text REtrieval Conference (TREC) [43].

Graded relevance judgements solve this limitation of binary relevance judgments, which provides multi-level relevance scales. Existing work on graded relevance judgements uses different relevance scales with different numbers of grades and different definitions of grades. Yang et al. [51] uses 3-point scale relevance judgements ("irrelevant", "fair", "relevant") to evaluate the quality of image results from a commercial search engine. 4-point scale relevance judgments ("perfectly relevant", " highly relevant", "relevant", "irrelevant") are applied in the deep learning track of TREC 2019-2021 [18], while web track in TREC 2012-2014 defined graded relevance score on a 6-point scale ("nav", "key", "hrel", "rel", "non", "junk") [17].

Google also provides search quality evaluator guidelines. The guidelines define the rating of web results as fully meets, highly meets, moderately meets, slightly meets, fails to meet [25], where fully meets means almost all users are fully and immediately satisfied with the result and there is no need to view other results, highly meets means the result is very helpful for many users and there might be need to view other results for some users, moderately meets means the result is helpful for some users and there might be need to view other results for many users, slightly meets means the result is weakly related to the query and more users still want to explore more, and fails to meet means the result completely fails to be related with the query.

A popular and standard graded relevance measure for both research and industry is normalized discounted cumulative gain (nDCG) [28]. It is considered as a primary evaluation measure in the recent TREC tasks. Another graded relevance measure to solve novelty and diversity is expected reciprocal rank (ERR) [12]. It is commonly used in the TREC web experiments considering the novelty and diversity of documents. The method will be explained more in the following sections.

Although graded judgements have been researched for a long time, they have several known limitations: 1) There are no universally adopted standards of relevance grades. As we mentioned above, different work uses different numbers of grades and different definitions of grades. It is hard to apply relevance grades consistently [9]. 2) Unlike binary relevance grades, assessors take more effort to precisely decide the multi-level relevance grades. When the number of grades level and other factors increases, it is more complex to label each item. 3) Relevance scores cannot be interpreted clearly as the level of user satisfaction. It is not clear that higher relevance scores represent better user experience [4].

2.2 NDCG

Normalized discounted cumulative gain (NDCG) is a measure to evaluate the performance of retrieval results and judge the relevance of retrieved documents[28]. Compared to previous evaluation measures, it supports the graded relevance which gives the degree of relevance of documents, while previous evaluation measures focus on binary relevance that only specify whether a document is relevant or not. NDCG also uses a discounted factor over the rank to devalue retrieved documents, while previous evaluation measures use uniform weights.[46]

NDCG is used widely as an evaluation metric on search engines. In the web search engine applications, graded relevance is often used in judging the relevance of web documents. NDCG fits this need and allows graded relevance. Web search engines also care about top ranking results, because documents ranked top usually obtain more attention than one ranked bottom. NDCG is able to weight documents with higher ranking greater, and consider them as more useful and relevant documents.

The usage of NDCG is not limited to evaluation, but also as a ranking metric. Ferrari et al.[19] used NDCG as a primary effectiveness measure for top-n recommendation tasks. Karmaker et al.[29] applied NDCG into a case study with learning-to-rank methods.

NDCG comes from discounted cumulative gain (DCG). A traditional formula of DCG at a particular rank position k is shown in equation 2.1, where rel_i is the graded relevance of the result at position i. It applies a discounted factor and accumulates the relevance score of retrieval documents in the ranked list. It is theoretically proved that DCG with logarithmic discount has consistent distinguishability, which means that for every pair of substantially different ranking functions, DCG can decide which one is better in a consistent manner[46].

$$DCG_k = \sum_{i=1}^{k} \frac{rel_i}{\log_2(i+1)} \tag{2.1}$$

A alternative formula of DCG commonly used in the industry is:

$$DCG_k = \sum_{i=1}^{k} \frac{2^{rel_i} - 1}{\log_2(i+1)} \tag{2.2}$$

NDCG at position k (equation 2.3)is DCG at position k divided by ideal discounted cumulative gain (IDCG).

$$NDCG_k = \frac{DCG_k}{IDCG_k} \tag{2.3}$$

Figure 2.1: Example documents with relevance scores.

IDCG is defined as equation 2.4, where REL_k is the list of relevance documents, sorted by their relevance score, up to position k. After ranking all relevant documents in the descending order based on their relevance score, IDCG is the DCG of the top k documents.

$$IDCG_k = \sum_{i=1}^{|REL_k|} \frac{2^{rel_i} - 1}{\log_2(i+1)} \tag{2.4}$$

A perfect ranking is when actual ranking is exactly the same as ideal ranking. In this case, the value of DCG and IDCG are the same, and NDCG will be 1.0. The value of NDCG ranges from 0.0 to 1.0.

As an illustrative example, consider the calculation of NDCG on the corpus in Figure 2.1. Suppose we have 7 documents in the corpus, and they are ordered by a ranking

node	i	rel_i	$\log_2(i+1)$	$\frac{rel_i}{\log_2(i+1)}$
B	1	3	1	3
A	2	4	1.585	2.524
H	3	4	2	2
D	4	2	2.322	0.861
G	5	1	2.585	0.387
C	6	2	2.807	0.713
F	7	1	3	0.333

Table 2.1: Calculate DCG for each position i in the example.

node	i	rel_i	$\log_2(i+1)$	$\frac{rel_i}{\log_2(i+1)}$
A	1	4	1	4
H	2	4	1.585	2.524
B	3	3	2	1.5
C	4	2	2.322	0.861
D	5	2	2.585	0.774
G	6	1	2.807	0.356
F	7	1	3	0.333

Table 2.2: Calculate IDCG for each position i in the example.

algorithm as the upper part of Figure 2.1. Each document has a relevance score on a scale of 0-4 with 0 meaning not relevant, 4 meaning highly relevant, and 1-3 meaning the degree of relevance. As shown in the figure, document B has a relevance score of 3, document A has a relevance score of 4, etc.

The DCG for each position i in order is illustrated in Table 2.1. So the DCG_7 of this ranking is:

$$DCG_7 = \sum_{i=1}^{7} \frac{rel_i}{\log_2(i+1)} = 3 + 2.524 + 2 + 0.861 + 0.387 + 0.713 + 0.333 = 9.818$$

One of the ideal ordering is shown in the middle part of Figure 2.1. For each position i, the IDCG is represented in Table 2.2, then $IDCG_7$ or DCG of this ideal ranking is

$$IDCG_7 = \sum_{i=1}^{|REL_7|} \frac{rel_i}{\log_2(i+1)} = 4 + 2.524 + 1.5 + 0.861 + 0.774 + 0.356 + 0.333 = 10.348$$

Finally, we can get NDCG:

$$NDCG_7 = \frac{DCG_7}{IDCG_7} = \frac{9.818}{10.348} = 0.949$$

There are two ideal rankings in the Figure 2.1. Although the order of documents are different, the order of relevance score remains same. NDCG is based on the relevance score at each position, so the values of NDCG of two ideal rankings are same.

Although NDCG has been developed for years and used widely in information retrieval applications, it still has several drawbacks. Firstly, NDCG needs careful and precise determination of the relevance score for each document. The standard of NDCG such as the level of relevance and its meanings have to be defined[9]. Human assessors also have to follow these definitions to determine the score consistently. While NDCG is applied in commercial contexts, human assessors have to put numerous effort into making relevance scores for tons of documents consistently[16].

Secondly, NDCG poorly accommodates other factors besides relevance when evaluating the ranking quality[14]. It assumes that documents with higher positions in the result list are more relevant and useful, while the importance of documents depends on the multiple factors in reality. For example, when the length of documents is a factor, suppose there are two documents containing the similar information, the shorter document is more likely to be preferred than the longer document. NDCG will not typically represent this information.

2.3 RBO

Webber et al. [47] define an indefinite rank similarity measure as a measure with three requirements for web search results. Firstly, this measure has to weigh differently by the position in rank lists. The documents at the top of rank lists are more important than those at the bottom, and these documents will obtain more attention. In this case, these top documents should be given higher weight than bottom documents by a rank similarity measure. Secondly, this measure should be able to handle incomplete rankings. Incomplete rankings mean these rankings are mutually non-conjoint, which means that a document shows up in one ranking, but not in other rankings. Thirdly, when calculating the rank similarity, this measure should be consistent at whatever positions in the ranking lists, instead of defining an arbitrary cutoff position.

Previous rank similarity measures fail to fulfill all of three criterias. Kendall's τ [32] is a widely-used rank correlation coefficient in the information retrieval domain, which is based on the number of concordant pairs of items in two rankings. However, it does not satisfy any one of indefinite rank similarity measure requirements[47]. Extending this work, Yilmaz et al.[53] developed a top-weighted variant τ_{AP} based on the average precision metric. It satisfies the requirements of top-weightedness, but not of incompleteness and indefiniteness.

With these three qualities of top-weightedness, incompleteness and indefiniteness, Webber et al. proposed rank-biased overlap (RBO) to compute indefinite rank similarity between rankings [47]. RBO is an overlap-based metric, and it basically biases the proportional overlap at each position in rank lists.

RBO can be applied to compute the similarity between actual ranking R and ideal ranking I. Let $R_{1:i}$ denote the items at rank i in R, and let $I_{1:i}$ denote the items at rank i in I. The intersection between R and I at rank i is:

$$Intersection = R_{1:i} \cap I_{1:i} \tag{2.5}$$

and the size of this intersection is the overlap of R and I at rank i:

$$Size = |Intersection| = |R_{1:i} \cap I_{1:i}| \tag{2.6}$$

The agreement between R and I at rank i is defined as the proportion of overlap and rank i:

$$Agreement = \frac{Size}{i} = \frac{|R_{1:i} \cap I_{1:i}|}{i} \tag{2.7}$$

Adding weight at rank i $(1-p)\cdot p^{i-1}$ into average agreement, the formula for RBO is

$$RBO(R,I) = (1-p)\sum_{i=1}^{\infty} p^{i-1}\frac{|R_{1:i}\cap I_{1:i}|}{i} \tag{2.8}$$

The value of RBO is ranged from 0.0 to 1.0, where 0 means actual ranking and ideal ranking are disjoint, 1 means these rankings are identical. The parameter $0 < p < 1$ represents users' persistence: suppose an user always starts with the first item in rankings, the user has probability p to continue to the next item in rankings and $1-p$ to stop. Larger p value means less top-weighted, and vice versa.

Webber et al.[47] show the strong correlation between RBO and traditional effectiveness measures. They used 10 relevant documents and 90 non-relevant documents, and randomly swapped relevant and nonrelevant documents 25 times. Everytime they swapped, they recorded similarity and average precision between the new ranking and ideal ranking. They found RBO has a close correlation with average precision, especially with p = 0.98.

Clarke et al.[16] explore the strong correlations between RBO between and actual and ideal ranking, and NDCG with a nearly linear relationship. They represent the value p of RBO and NDCG@k are closely rank correlated. When $p = 0.85$, RBO can be matched NDCG@5; when $p = 0.9$, RBO can be matched NDCG@10; and when $p = 0.95$, RBO can be matched NDCG@20.

As an illustrative example, consider the calculation of RBO on the rankings in Figure 2.1. Suppose we have 7 documents in the corpus, they are ordered by ranking algorithm as upper part of Figure 2.1 and they have two ideal rankings as middle and lower part of Figure 2.1. For this example, we will use $p = 0.95, DEPTH = 7$.

In order to calculate RBO for actual ranking and ideal ranking 1, we need to calculate the agreement at each depth i, shown in Table 2.3. Then we can get RBO:

$$\begin{aligned} RBO(R,I1) &= (1-p)\sum_{i=1}^{\infty} p^{i-1}\frac{|R_{1:i}\cap I_{1:i}|}{i} \\ &= (1-0.95)\cdot 4.18175 \\ &= 0.209 \end{aligned}$$

The agreement at each depth i between actual ranking and ideal ranking 2 in the figure 2.1 is shown in the table 2.4. So the RBO between actual ranking and ideal ranking 2 can be calculated as following:

$$\begin{aligned} RBO(R,I2) &= (1-0.95)\cdot 2.287 \\ &= 0.457 \end{aligned}$$

i	R_i	I_i	$R_{1:i} \cap I_{1:i}$	$\frac{\lvert R_{1:i} \cap I_{1:i} \rvert}{i}$	p^{i-1}
1	B	A	0	0	1
2	A	H	1	0.5	0.95
3	H	B	3	1	0.903
4	D	C	3	0.75	0.857
5	G	D	4	0.8	0.815
6	C	G	6	1	0.774
7	F	F	7	1	0.735

Table 2.3: Calculate agreement at each depth i between actual ranking and ideal ranking 1 in the example.

i	R_i	I_i	$R_{1:i} \cap I_{1:i}$	$\frac{\lvert R_{1:i} \cap I_{1:i} \rvert}{i}$	p^{i-1}
1	B	H	0	0	1
2	A	A	1	0.5	0.95
3	H	B	3	1	0.903
4	D	C	3	0.75	0.857
5	G	D	4	0.8	0.815
6	C	F	5	0.833	0.774
7	F	G	7	1	0.735

Table 2.4: Calculate agreement at each depth i between actual ranking and ideal ranking 2 in the example.

Unlike NDCG, RBO does not require relevance scores, and the position of each item in the ranking affects the value of RBO. From the above example, we can see that when calculating the rank similarity between different ideal rankings and actual ranking, RBO scores are different.

2.4 Preference Judgements

Preference judgements have been a good alternative to graded judgements for a long history. As far as 1990, Rorvig [37] shows that preference can be a substitute for purely relevance scores when evaluating a sensitive retrieval system. Preference can identify the difference between equally relevant documents and provide better reflections on finding highly-relevant documents. Rovig also points out that it is costly to use preference judgements to judge fully a list of items, which requires heavier effort than using graded judgements. Frei and Schauble [24] propose a new effectiveness measure for relative judgements and suggest that it is easier and more consistent for human assessors to give relative judgements than graded judgements. Yao [52] studies the user preference on documents and retrieval results and proposes a new evaluation measure based on the distance between actual ranking from user preference judgements and ideal ranking from systems.

Later work on preference judgements, Carterette et al. [7, 9, 8] make great progress on applying preference judgements practically to evaluate search engines. Carterette et al. [7] construct one of earliest test collections of preference judgements and prove that preferences perform better than binary labels for some evaluation measures, although the difference is small. Carterette et al. [9] give the definition of transitivity of preference judgments: if documents i is preferred to document j and document j is preferred to document k, the assessor also prefers document i to document k. They also show that there is no need to compare all pairs of documents, and assessors spend less time and obtain better agreement per preference judgement than graded judgement. More importantly, they introduce some preference evaluation measures: ***ppref*** (precision of preferences), ***wpref*** (weighted precision of preferences), ***rpref*** (recall of preferences) and average precision of preferences [8, 9].

Based on Carterette's work, Clarke et al. [16, 14, 15] focus on judging partial preferences and identifying top-k items, instead of comparing all pairs of items and giving a full rank list. Extending Carterette's work on preference measures ***ppref*** and ***wpref***, Sakai and Zeng [40] define a variety of preference-based measures and investigate the relationship between these measures and users' search engine result pages (SERP) preferences.

More recently, preference judgements are not restricted to only documents in web search. Xie et al. [49] are the first one to apply preference judgement on web image search. Unlike web pages that display in a sequential list, web images are placed in a grid-based style. They combine grid-based assumptions and previous preference measures defined by Carterette et al. [9], and propose a preference-based evaluation metric on web image search.

Instead of defining relevance grades and assigning relevance labels to each item in the result, preference judgements is that assessors look at two items A and B, and prefer item A over another item B, which can be expressed as pair (A,B). While there are other ways of deciding the preferences, such as relevance scores, clicks and signals, these are beyond the scope of present study. Preference judgments are also considered as a binary decision in this paper, while some studies focus on capturing various other factors and doing more complex preference judgements such as [11, 10]. Preference judgments are context-free in this paper, which means that there are no restrictions on redundancies and conflicts, where redundancies represent that pair (A.B) exists multiple times and conflicts represent that pair (A,B) and pair (B,A) coexist in a data collection.

Compared to graded judgements, preference judgements have several advantages: 1) There is no need to determine the number of grades and definition of each grade like graded judgements. Preference judgements only ask assessors for preferences over two items. 2) It is easier and less time consuming for assessors to decide the preference over two items than assigning relevance score to an item in the search result pool [9].

2.5 Clarke et al. Work

Using preference judgements brings two problems: 1) Under the assumption of transitivity of preference judgements (if documents i is preferred to document j and document j is preferred to document k, the assessor also prefers document i to document k), suppose we have a pool with n number of documents, assessors need to judge $O(nlogn)$ pairs of documents to give a full rank of documents. If transitivity is not assumed, then assessors take $O(n^2)$ preference judgments to rank all documents. Compared to graded judgements that only take n time to assign relevance grades, preference judgements take more effort providing a rank of all documents. 2) Graded judgements have multiple evaluation measures such as NDCG, ERR and RBP, while there haven't been an evaluation metric commonly used in preference judgements.

Clarke et al. [15] addresses these two problems by using partial preferences and proposing an evaluation measure, called compatibility, to evaluate a system's performance by

measuring maximum similarity between an actual ranking generated by the system and an ideal ranking.

Based on the previous work of Clarke and his colleagues [16], partial preferences are defined as any weak ordering of items from data collection that is able to create a preferred ranking for a query, where ties between items are allowed. This weak ordering can be derived from relevance grades, pairwise comparisons or both. In order to minimize the effort, Clarke et al. used partial preferences to find top-k items for a query, where k is less than or equal 5. Their approach is divided into several steps:

- They initially assess the graded relevance of all items in the pool, which filters low quality items or non-relevant items and decreases the preference judgements between these items. This step results in a reduced and high quality candidate pool C, where $|C| \leq k$.
- To reduce the size of the candidate pool, they put candidates into a elimination tournament judged by crowdsourced workers: if the size of candidate pool is greater than threshold F, where $|C| > F > k$, each candidate will be randomly paired with P or $P + 1$ other candidates, where $F > P > k$. When a candidate is losing in a majority of pairings, this candidate is eliminated. This elimination is repeated until the size of the candidate pool is less than or equal to F.
- To determine the final order, all remaining candidates are paired with all other remaining candidates. These pairings are judged by crowdsourced workers. Based on the number of pairs the candidates win, all remaining candidates are ranked and cut to the top-k. If ties exist at rank k, the candidates with tied scores will be kept in the ranking.

Evaluating a system's performance is to compute the maximum similarity between an actual ranking generated by a ranker for a query and a set of ideal rankings, which they call this as the compatibility of the actual ranking and a set of ideal rankings. There are three steps to compute the compatibility:

- Finding an actual ranking generated by a search engine.
- Defining a set of ideal rankings for a query.
- Choosing the rank similarity measure to compare rankings.

In the work of Clarke et al. [15], actual ranking is generated from the tournament and they give the definitions of ideal rankings. Ideal rankings for a query are defined as a set of equivalence classes, or "effectiveness levels". Ideal rankings for a query are defined as a set of equivalence classes, or "effectiveness levels". Effectiveness levels are defined as a set of $L_1, L_2, ..., L_T$, where $L_1 < L_2 < ... < L_T$ and L_T are the top level. An extra level L_0 includes the items that are not in any of the other effectiveness levels. Each effectiveness level contains one or more items in the pool, and the number of effectiveness levels can be varied from query to query. For graded relevance, effectiveness levels are exactly matched to graded relevance score, where L_0 represents non-relevance. For an ideal ranking containing top-k items, effectiveness levels correspond to this ideal ranking, where the top item is in L_k, the second item is in L_{k-1}, and so on.

Once ideal rankings and actual ranking are defined, they calculate the compatibility by using RBO to compute the maximum similarity between ideal ranking and actual ranking.

We will extend the work of Clarke and his colleagues, adapt compatibility into preference graph applications. We will introduce our algorithm in the next chapter.

Chapter 3

Preference Graph Compatibility

3.1 Define Ideal Ranking

Based on the work of Clarke et al. [15], we follow the definition of compatibility (the maximum similarity between an actual ranking generated by a ranker for a query and a set of ideal rankings) and how they compute the compatibility of the actual ranking and a set of ideal rankings. There are three requirements to compute compatibility: 1) define a set of ideal rankings for a query that are returned by a system as a perfect result; 2) define an actual ranking generated by a system that is used to compare with ideal rankings; 3) choose a suitable rank similarity measure to compare the rankings between actual ranking and ideal rankings. We apply these definitions to preference graph applications.

We firstly introduce some notations for the preference graph. We consider preferences between items as a directed graph $G = (V, E)$, where a vertice represents an item in the search result and an edge (u, v) represents item u is more preferred than item v. Graph G can be cyclic, for example, the edge (u, v) and (v, u) can coexist, which means that the preferences between item u and item v are contradicted. The edges in graph G can be repeated, for example, multiple edges in E pointing from u to v can coexist, which means that the preferences between item u and item v are judged independently multiple times.

Given a preference graph G with n vertices and m edges, we define a set of ideal rankings as following:

- Any directed acyclic graph(DAG) that is resulting from removing minimal feedback arc sets of the preference graph G;

- Any sequence of the vertices $v_1, v_2, \ldots, v_n$ that is consistent with this DAG, where the order of elements in this sequence does not contradict with this DAG;
- The ideal ranking is the sequence that is closest to the actual ranking.

In the ideal situation, in order to find the ideal rankings, firstly we have to build a set of DAGs by removing the minimal number of edges from preference graph G. Then we have to build a set of chains that are aligned with each DAGs. Finally we decide a set of ideal ranking by finding the chains that are closest to the preference graph G, where an ideal ranking contradicts the smallest number of preferences.

However, this ideal and theoretical solution is not plausible (being NP-Complete). Each step requires large computations. We will talk about each step in detail.

3.2 Minimal FAS Problem

Computing a feedback arc set(FAS) is an advantageous way to generate a DAG that closely represents a large and complex network with minimal destruction of the original structure of this directed graph. A feedback arc set or feedback edge set of a directed graph G is a subset of arcs or edges F of G such that removing F from G leaves an acyclic graph. In other words, it is also a set that contains at least one edge of every cycle in G. A minimal feedback arc set of G is a subset of arcs F with minimal cardinality. In other words, removing as few feedback edges as possible from G to generate a DAG, these edges are called a minimal feedback arc set.

The computation of a minimal feedback arc set is called a minimal feedback arc set problem, which is known as a hard computational problem. The decision version of this problem is, given a directed graph G, can all cycles in G be broken by removing at most k edges? This is one of Karp's 21 NP-complete problems [30], whose NP-hardness follows the reduction from vertex cover problem.

3.2.1 Approximate FAS Alogorithms

An approximate algorithm is widely used in combinatorial optimization problems that are seeking to find the best solution from all feasible solutions, or find an nearly optimal solution quickly instead of finding an exactly optimal solution. For such problems, approximation algorithms find an approximate solution that is proved to be closely optimal with a guaranteed factor α.

Vazirani defines approximation algorithms in his book [45]: an algorithm A is said to be a factor α approximation algorithm for an optimization problem P iff on every instance I of the problem P, A produces a feasible solution s for I such that $f_P(I,s) \leq \alpha(|I|)\dot{O}PT(I)$ or $f_P(I,s) \geq \alpha(|I|)\dot{O}PT(I)$, where $OPT(I)$ denotes the objective function value of an optimal solution to instance I. It means that this algorithm finds a solution s within a factor α of the optimal solution for every instance I of the problem P. If the problem is a minimization problem, then $\alpha > 1$ and it implies that the value of the approximate solution is at most α times the optimal solution. If the problem is a maximization problem, then $\alpha < 1$ and it implies that the value of the approximate solution is at least α times the optimal solution.

Approximation algorithms are useful and efficient for NP-Hard problems that do not exist polynomial time solutions unless $P = NP$. The runtime of an approximation algorithm is bounded polynomially by the instance I, $poly(I)$. Approximate algorithms are applied to solve lots of NP-hard problems, such as set cover problem, traveling salesman problem, multiway cut problem, feedback arc set problem and etc.

The FAS problem has been studied for decades in mathematical programming[26, 36] and approximation algorithms [3, 5, 6, 21, 22, 34, 35]. The first approximation algorithm for the FAS problem was given by Leighton and Rao [35]. Their approximation factor is $O(\log^2 n)$ in the unweighted case, where n is the number of vertices of the input graph. They use an $O(\log n)$ approximation algorithm for balanced cuts that separates the graph into two parts with approximately equal size, which they call "quotient cuts". They prove this solution is a polylog-times optimal approximation algorithm. Later, Klein et al. [34] improve this algorithm to an $O(m^2 \log m)$ expected-time randomized algorithm. Even et al. [22] provide an approximation algorithm solution with an $O(\log n \log\log n)$ approximation factor and $O(n^2 M(n) \log^2 n)$runtime, where $M(n)$ represents the complexity of matrix multiplication. This solution is currently the best known approximation algorithm due to the best approximation ratio for the FAS problem.

There are some works focusing on various heuristics for computing an approximate minimum feedback arc set. Eades et al. [21] provides a greedy algorithm to generate a vertex sequence with approximately minimal feedback edge in $O(m)$ runtime, where m denotes the number of edges of the input graph. Berger and Shor [5] develop a 2-approximation algorithm with $O(m+n)$ runtime for the maximum acyclic subgraph problem. This algorithm returns the maximum acyclic subgraph G' of the input graph G so that the arcs in G but not in G' are exactly the same as the minimal FAS of G. Ailon et al.[3] present a 3-approximation algorithm based on classical sorting algorithms for the FAS tournament problem, and later Brandenburg and Hanauer [6] extend it to heuristics for the FAS problem, with the runtime complexity of $O(n \log n)$.

Algorithm 1 GREEDY FAS

Input
    Directed graph $G = (V, E)$
Output
    Linear arrangement A
$s_1 \leftarrow \emptyset$, $s_2 \leftarrow \emptyset$
while $G \neq \emptyset$ **do**
    while G contains a sink **do**
        choose an arbitrary sink u
        $s_2 \leftarrow us_2$
        $G \leftarrow G \setminus u$
    while G contains a source **do**
        choose an arbitrary source u
        $s_1 \leftarrow s_1u$
        $G \leftarrow G \setminus u$
    choose an arbitrary vertex u for which $\delta(u)$ is a maximum
    $s_1 \leftarrow s_1u$
    $G \leftarrow G \setminus u$
return $A = s_1 + s_2$

Figure 3.1: Greedy FAS algorithm.

Among multiple algorithms of minimal FAS problem, greedy algorithm [21] has an excellent performance overall. Simpson et al. [42] analyze the performance and efficiency of a wide range of algorithms, and find that the greedy algorithm provides better balance between scalability and quality, which always produce the smallest or the second smallest FAS size while only requiring a linear time. In addition, the output of greedy algorithm is a sequence of vertices with approximately minimal backward arcs, which satisfies our requirements of ideal rankings. Therefore, greedy algorithm of minimal FAS problem is chosen in this paper to generate ideal rankings.

3.2.2 Greedy FAS

Greedy algorithm for minimal FAS problem (Greedy FAS) was developed by Eade et al. [21] as a fast and effective approximation algorithm. Instead of computing minimal FAS directly, Greedy FAS outputs a vertex sequence with minimal backward arcs. Suppose the vertices of G are ordered on a horizontal line and labelled $v_1, v_2, \ldots, v_n$, such arrangement is called a vertex sequence and denoted as $s = v_1 v_2 \ldots v_n$. Each vertex sequence is a FAS with all the leftward arcs $v_i v_j$, such that $j > i$. In the inverse situation, each FAS might imply some vertex sequences. Therefore, Greedy FAS converts FAS problem into the problem of determining a vertex sequence s' such that the FAS of s' is equivalent to the minimal FAS of G.

For each vertex $u \in V$, let $d^+(u)$ denote the out-degree of u, $d^-(u)$ denote the in-degree of u. At each iteration of Greedy FAS, the algorithm removes sinks and sources from input graph G, and then removes a single vertex u for which $\delta(u) = d^+(u) - d^-(u)$ is currently a maximum. If a vertex u removed from G is a sink, it will be prepended to a vertex sequence s_2; otherwise, it will be appended to a vertex sequence s_1. When all the vertices are removed from G and G becomes an empty graph, the output vertex sequence s is computed by the combination of s_1 and s_2, which is a linear arrangement of vertex whose backward arcs are equivalent to the minimal FAS of G. The pseudocode for Greedy FAS is presented in Algorithm 1.

Greedy FAS greedily removes sinks, sources, or a vertex with maximum δ to generate a final vertex sequence. In other words, this algorithm greedily moves all the sinks to the left side of final vertex sequence, moves all the sources to the right side of final vertex sequence, and moves one maximum δ vertex to the right side of final vertex sequence. It minimizes the number of backward edges (pointing from right to left) in the final vertex sequence.

As an illustrative example, consider the implementation of Greedy FAS on the graph in Figure 3.2. There are 7 vertices, A,B,C,D,F,G,H, and 7 edges in this directed cyclic graph G. Greedy FAS executes on this graph as following steps:

1. Initially, there are 3 sinks in this graph, vertex D, vertex F, and vertex G. We choose an arbitrary vertex from these three sinks, remove this vertex and its arcs from graph G and prepend to s_2. We repeat this process until vertex D, vertex F and vertex G are all removed from graph G. Suppose the order we remove vertices is D, F, G, then current s_2 = [G, F, D].

2. Next, we process the sources in this graph, which are vertex A and vertex H. We choose an arbitrary vertex from these two sources, remove this source and its arcs

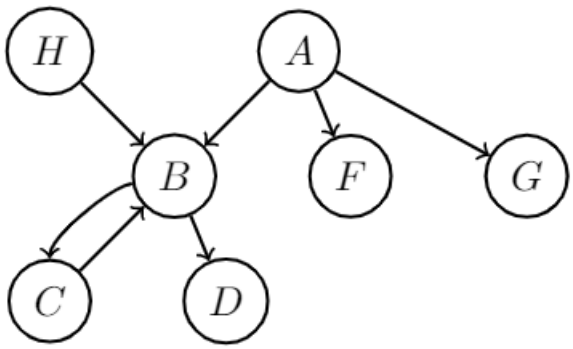


Figure 3.2: An example of cyclic directed graph.

from graph G and append to s_1. This process is repeated until both vertices A and H are removed from this graph G. Suppose we remove vertex A firstly and then vertex H, then current s_1 = [A, H].

3. After that, we remove the vertex u with maximum $\delta(u)$, Current graph G only has 2 vertices, B and C, and 2 cyclic edges between them. $\delta(B) = 1 - 1 = 0, \delta(C) = 1 - 1 = 0$. Since vertex B and C have the same and maximum δ value, we choose an arbitrary vertex. Suppose we choose vertex B, vertex B and its arcs are removed from graph G, and appended to s_1, then current s_1 = [A, H, B].

4. After removing vertex B, a single vertex C without any edge is left. As such, vertex C is removed from graph G and prepended to s_2, so s_2 = [C, G, F, D].

5. Graph G becomes an empty graph, the iteration stops, and the final vertex sequence $s = s_1 + s_2$ = [A, H, B, C, G, F, D]. In our final vertex sequence, there is only one backward arc from vertex C to vertex B, which represents the solution of this FAS problem is the single arc from vertex C to vertex B.

An interesting property of Greedy FAS is arbitrariness. When there are multiple vertices satisfying requirements at each step, this algorithm always makes an arbitrary decision. If the order of adding vertices to sub-sequences is different, then the sub-sequences will be different and the linear arrangement output of this algorithm will have multiple possible solutions.

The example mentioned above illustrates only one possible output of Greedy FAS, while there are several possible formations of sequences at each step. There are 6 possible arrangements for s_2 at step 1: any combinations of vertex D, F, G. At step 2, s_1 has 2 possibilities: any combinations of vertex A and H. If vertex C is chosen at step 3, then vertex C will be appended to s_1, and vertex B will be prepended to s_2 at step 4. In this case, there are still two possible sequences for s_1 and 6 possible sequences for s_2. With all combinations of s_1 and s_2, there are 12 different possible linear arrangements as the output of Greedy FAS algorithm, although they all refer to one solution of this FAS problem, the arc from vertex B to vertex C. Similarly, if vertex B is chosen at step 3, then there are 2 possible sequences for s_1, 6 possible sequences for s_2, and 12 different possible outputs of Greedy FAS, while there is only single solution of this FAS problem, the arc from vertex C to vertex B. Therefore, with arbitrary decisions in Greedy FAS, there are 24 possible vertex sequences in total for this FAS problem.

3.3 DAG Alignment Problem

Finding an aligned chain with a DAG is a NP-complete problem. Falconer et al. [23] proves that the weighted alignment between two DAGs is a NP-complete problem. The decision version of this problem is NP-complete, shown by reducing from the 3SAT problem. Although they do not explicitly show the alignment between a sequence and a DAG is NP-complete, they do define a chain C_n as a DAG with n vertices $v_1, v_2, \ldots, v_n$ and directed edges $(v_1, v_2), (v_2, v_3), \ldots, (v_{n-1}, v_n)$, which implicitly means that DAG and chain alignment is also NP-complete. They also represent that a solution of the DAG alignment problem is polynomial time complexity.

Finding just one alignment between the actual ranking and a DAG requires polynomial time with the number of items in the search list, while finding all alignments between actual ranking and a set of DAGs is much more difficult and time consuming.

3.4 NP-Completeness of Ideal Solution

Producing one DAG by dropping a minimal feedback arc set from preference graph G is NP-complete, while we have to find all the DAGs that satisfy this requirement, which requires more effort computing. Generating one sequence that is consistent with a DAG is also NP-complete, while finding all the sequences that are consistent with each DAG we find before is more difficult and needs numerous computations, and these sequences will

Algorithm 2 GreedyPGC

Input
    Directed graph $G = (V, E)$, actual ranking R
Output
    Preference Graph Compatibility
$s_1 \leftarrow \emptyset$, $s_2 \leftarrow \emptyset$
while $G \neq \emptyset$ **do**
    while G contains a sink **do**
        $u \leftarrow$ RankSink($\forall$ *sinks* $\in G, R$)
        $s_2 \leftarrow us_2$
        $G \leftarrow G \setminus$u
    while G contains a source **do**
        $u \leftarrow$ RankSource($\forall$ *sources* $\in G, R$)
        $s_1 \leftarrow s_1u$
        $G \leftarrow G \setminus$u
    $u \leftarrow$ RankSource($\forall\ v \in V$ with maximum $\delta(v), R$)
    $s_1 \leftarrow s_1u$
    $G \leftarrow G \setminus$u
$I \leftarrow s_1 + s_2$
return RBO(I, R)

Figure 3.3: Greedy PGC algorithm.

be a large set. From this large set of sequences, we still have to choose the best alignments that are closest to the preference graph G. It is not a simple task and hard to precisely find the closest chains from a huge set of alignments.

Therefore, we decide to choose a faster method, an approximation algorithm, for computing approximations to such alignments, and finding a set of ideal rankings becomes plausible.

Algorithm 3 RankSink

Input
    Node list N, actual ranking R
Output
    Node n with the lowest actual ranking
$p \leftarrow \emptyset$
for all node $n_i \in N$ **do**
    if n_i exists in R **then**
        $rank \leftarrow$ rank of n_i in R
    else
        $rank \leftarrow \infty$
    $p \leftarrow p + (n_i, rank)$
return n with $max(rank)$ in p

Algorithm 4 RankSource

Input
    Node list N, actual ranking R
Output
    Node n with the highest actual ranking
$p \leftarrow \emptyset$
for all node $n_i \in N$ **do**
    if n_i exists in R **then**
        $rank \leftarrow$ rank of n_i in R
    else
        $rank \leftarrow \infty$
    $p \leftarrow p + (n_i, rank)$
return n with $min(rank)$ in p

Figure 3.4: Functions RankSink and RankSource in Greedy PGC algorithm.

3.5 Greedy Preference Graph Compatibility

In order to compute compatibility, we need to obtain the actual ranking, find the ideal ranking that is closest to the actual ranking, and use a suitable ranking similarity measure to compare these two rankings.

Given the actual ranking, the ideal ranking should have the least number of discordant pairs with the actual ranking. Discordant pairs mean the conflict of relationship between items in the different ranking, for example, if we have item A ranked higher than item B in the actual ranking, denoted as pair(A,B) in the actual ranking, but item A ranked lower than item B in the ideal ranking, denoted as pair(B,A) in the ideal ranking, then these two pairs are discordant. Greedy FAS algorithm generates a vertex sequence that minimizes the number of backward arcs, which satisfies the requirement of minimal discordant pairs. To fully satisfy the requirement of minimizing the number of discordant pairs between ideal ranking and actual ranking, we revise Greedy FAS algorithm to suit our needs.

Greedy Preference Graph Compatibility (Greedy PGC) is based on Greedy FAS to evaluate the performance of a system, which computes the preference graph compatibility. It takes a preference graph G and an actual ranking R as input, finds the ideal ranking I closest to the input actual ranking R, and outputs the compatibility of the actual ranking and the ideal ranking. The pseudocode for Greedy PGC is presented in Algorithm 2.

Unlike making an arbitrary decision when choosing the next vertex u in Greedy FAS, Greedy PGC calls functions, RankSink and RankSource, to determine the next vertex u based on the actual ranking R.

Function RankSink takes all sinks in the current graph G, denoted as $\forall\ sinks \in G$. If there are any sinks that do not exist in the actual ranking R, return an arbitrary one n; if all the sinks are in the actual ranking R, return the sink with lowest actual ranking R. The nodes that are not in the actual ranking R have a higher priority than other nodes. The pseudocode for RankSink is presented in Algorithm 3.

Function RankSource takes all sources, denoted as $\forall\ sources \in G$, or the vertices with maximum value of δ, denoted as $\forall\ v \in V$ with maximum $\delta(v)$. It chooses the vertex ranked highest in the actual ranking R or an arbitrary vertex that is not in the actual ranking R. The nodes existing in the actual ranking R have a higher priority to be chosen than other nodes. The pseudocode for RankSource is presented in Algorithm 4.

Besides adding above functions, we also change the output of algorithm. Greedy FAS generates a vertex sequence, while Greedy PGC has one more step: using RBO to calculate the similarity between actual ranking R and ideal ranking I, denoted as $RBO(I, R)$.

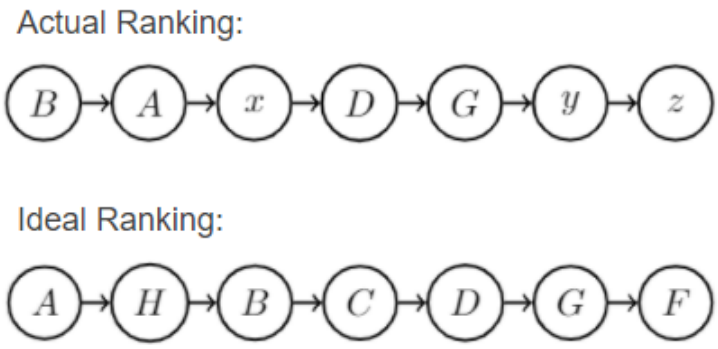


Figure 3.5: Example input and output of Greedy PGC.

As an illustrative example, consider the implementation of Greedy PGC on the preference graph in Figure 3.2 and the actual ranking in Figure 3.5. Suppose we have 7 items in the preference graph, and their actual rankings are ranked as the upper part of Figure 3.5. Greedy PGC execute on this graph and actual ranking as following steps:

1. Initially, there are 3 sinks in this graph, vertex D, vertex F, and vertex G. In the first inner loop of the algorithm, the order of prepending to s_2 is vertex F, vertex G and then vertex D, because vertex F is not in the actual ranking and vertex G is ranked lower than vertex D. Current s_2 = [D, G, F].

2. There are 2 sources in the current graph, and vertex A appears in the actual ranking while vertex H does not. In the second inner loop of the algorithm, the order of appending to s_1 is vertex A, and then vertex H. Current s_1 = [A,H].

3. Vertex B and C are left in the current graph and vertex with maximum δ value should be removed. $\delta(B) = 1 - 1 = 0, \delta(C) = 1 - 1 = 0$. Since vertex B and C have the same and maximum δ value, we call the function RankSource and choose vertex B to append to s_1 because vertex B exists in the actual ranking while vertex C does not. Current s_1 = [A, H, B].

4. After removing vertex B, a single vertex C without any edge is left. As such, in the second iteration of the outer loop, vertex C is removed from graph G and prepended to s_2. Current s_2 = [C, D, G, F].

5. Graph G becomes empty, and the iteration stops. Ideal ranking is the concentration of s_1 and s_2: $I = s_1 + s_2 =$ [A, H, B, C, D, G, F]. It is shown as the lower part of Figure 3.5.

6. RBO between ideal ranking I and actual ranking R is calculated as following, where $p = 0.95, DEPTH = 7$:

$$\begin{aligned} RBO(I, R) &= (1 - 0.95) \cdot 2.929 \\ &= 0.146 \end{aligned}$$

Finally, the function Greedy PGC returns the value of RBO.

Chapter 4

Experimental Comparisons

In the following chapters we experimentally compare PGC to recent efforts to make preferences and preference judgments usable in practice. We base our comparison on three factors used in the original papers, where these factors are discussed in detail [14, 15, 16, 39, 40].

1. The *agreement* of measures with SERP preferences, where pairs of SERPs are compared side-by-side by searchers or assessors. Agreement is expressed as a count of pairs where an evaluation measure agrees or disagrees with SERP preferences.

2. The *sensitivity* of measures in identifying significant difference between rankers. We determine sensitivity over a collection of rankers producing actual rankings for a set of queries by computing pairwise significance tests between them over the query set. Sensitivity is the proportion of pairs that show pairwise significant differences. For the pairwise test, we use a paired t-test with $p < 0.05$.

3. The *consistency* of measures as indicated by Kendall's τ correlation coefficient and illustrated by scatter plots. If we are proposing to replace one measure by another they should behave in similar ways, unless we can justify the differences.

4.1 Sakai and Zeng

SERP preferences provides a clear and natural basis for validating offline evaluation measures. For this reason, Sakai and Zeng [39, 40] employ agreement with SERP preferences

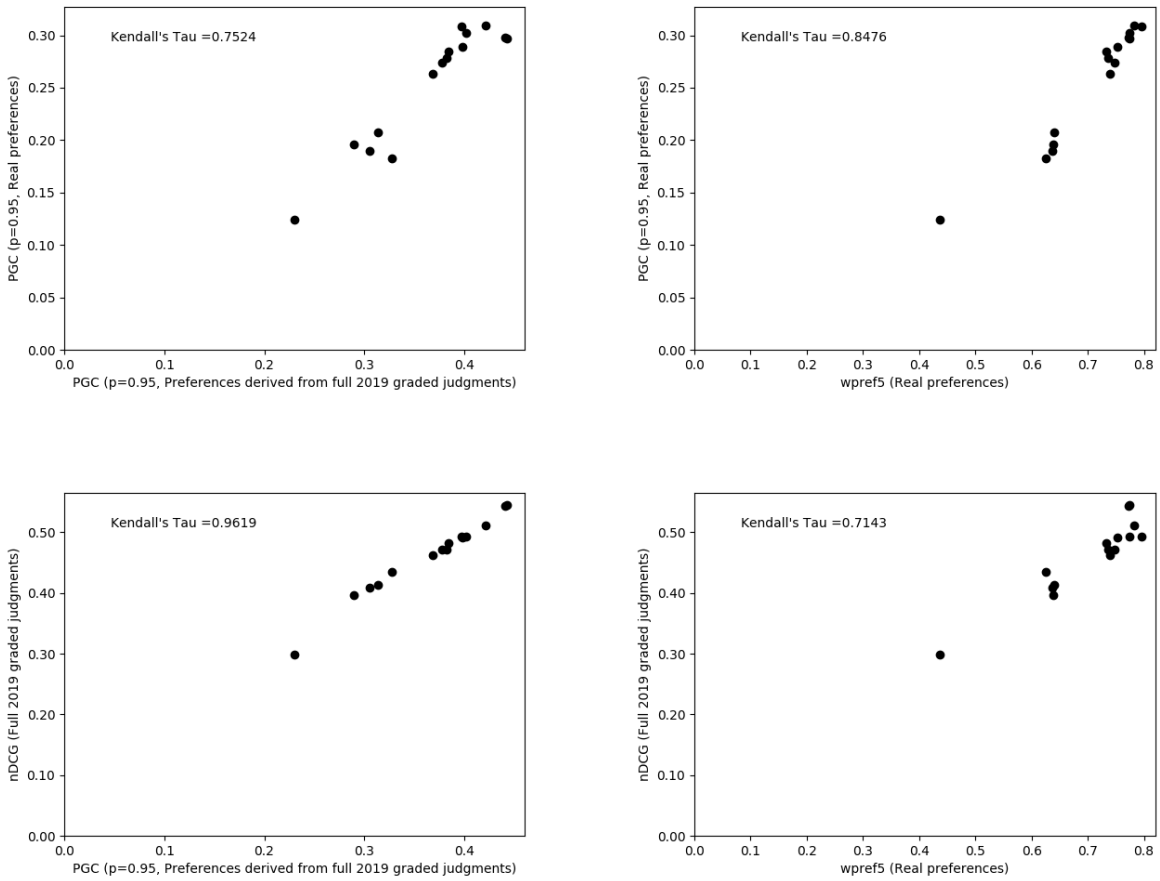


Figure 4.1: Scatter plots comparing measures from Sakai and Zeng [39, 40] with PGC. Each point is an experimental run submitted to the NTCIR-9 INTENT task. Measures are averaged over 43 queries. For PGC, $p = 0.95$.

Measure	Judgments	Agree	Disagree	Sensitivity
nDCG	Graded 2020	809	85	58.1%
nDCG	Graded 2019	809	85	48.6%
PGC	**Derived 2020**	**802**	**92**	**64.8%**
PGC	**Derived 2019**	**801**	**93**	**55.2%**
wpref6	Derived 2019	793	101	57.1%
PGC	**Combined**	**790**	**104**	**50.4%**
wpref5	Real	785	109	56.2%
PGC	**Real**	**778**	**116**	**59.0%**
ERR	Graded 2020	718	176	61.9%
ERR	Graded 2019	718	176	42.9%

Table 4.1: Agreement with SERP preferences and sensitivity for selected measures from Sakai and Zeng [40], along with PGC values (bolded).

as a key tool in their work. Since offline evaluation measures typically combine individual judgments into a summary score, SERP preferences can provide a direct indication of the success of these summaries.

Sakai and Zeng [40] review and extend past efforts to base evaluation measures purely on preference judgments. They define two large families of measures, which they called the "pref measures" and the "Δ-measures". Like PGC, the pref measures assume a preference multigraph, although Sakai and Zeng do not explicitly view preferences as a graph, instead referring to a "bag" of preferences. These pref measures are based on ideas originally proposed by Carterette et al. [8, 9]. Given a bag of preferences, the pref measures reflect the degree to which an actual ranking contradicts these preferences. Sakai and Zeng [40] define a family of pref measures (27 in total) that explore a variety of methods for computing these contradictions and weighting them according to their position in the actual ranking.

The Δ-measures are based on the Δ value defined in [2], which is a difference between the in-degree and out-degree of a vertex in the directed multigraph defined by a collection of preferences. This value is essentially the negative of the $\delta(v)$ function from Algorithm 2. Sakai and Zeng[40] map this Δ value into an nDCG-like gain value for use in traditional offline evaluation measures. They define a family of Δ-measures (eight in total) that replace the gain values in traditional measures, including nDCG and ERR, giving measures like Δ-nDCG, Δ-ERR, etc.

Sakai and Zeng [39, 40] base their work on a test collection developed as part of the INTENT task of the NTCIR-9 evaluation effort [38]. Like most academic test collections, this test collection consists of a corpus, queries, and judgments. The corpus comprises

Japanese-language documents taken from the ClueWeb09 Web crawl, which has been used extensively for academic research over the past decade[1]. For the NTCIR-9 INTENT task, the organizers created 100 queries over this corpus. Task participants submitted 15 experimental runs to the task. These runs were pooled and judged according to diversity criteria defined as part of the task. To explore evaluation measures for *ad hoc* search, as well as diversity, Sakai and Zeng [39] later converted these judgments to relevance grades on a three-point scale.

To study agreement of evaluation measures with SERP preferences, Sakai and Zeng [39] hired 15 assessors to judge SERP-SERP pairs, directly comparing the ranked top-10 documents from one run to those of another on a particular query. This effort produced 1,127 query-SERP-SERP triples. Of these triples, they selected 894 triples as a "high agreement" subset, which included only 43 of the original queries. Sakai and Zeng [40] employ these 43 queries and 894 query-SERP-SERP triples to study their family of 27 pref measures and 8 Δ-measures. The agreement of evaluation measures with these 894 triples is a key factor in their experimental study.

For the 43 queries they collected 119,646 triples (query-document-document) each of which indicates that an assessor preferred one document over the other for that query. These query-document-document triples contain redundancies and duplicates, forming a preference multigraph under our definition. In the remainder of this section, we extend the work of Sakai and Zeng [40] to our PGC measure.

Since we wish to explore a variety of preference graphs, as well as PGC. We employ four different preference graphs based on Sakai and Zeng [40], as defined below, including some not explored in that paper:

- **Real** — Preference graphs from the 119,646 query-document-document triples collected from assessors making side-by-side preference judgments. Sakai and Zeng [40] call these $PREF^{real}$. On average they contain 2,782 edges per query.
- **Derived 2019** — Preference graphs derived from graded judgments used in Sakai and Zeng [39]. On average they contain 1,808 edges per query.
- **Derived 2020** — Preference graphs deprived from graded judgments used in Sakai and Zeng [40], who filtered the 2019 graded judgments to keep only those appearing in the 894 query-SERP-SERP triples. Sakai and Zeng [40] call these $PREF^{derived}$. On average they contain 402 edges per query.

[1] http://lemurproject.org/clueweb09

- **Combined** — The union of Real and Derived 2019, i.e, all preferences available. We do not include Derived 2020 in Combined since it's already a subset of Derived 2019. On average these preference graphs contain 4,590 edges per query.

We call the set of graded judgments used in Sakai and Zeng [39] "**Graded 2019**" and the set of graded judgments used in Sakai and Zeng [40] "**Graded 2020**". Filtering these judgments might have minimal impact on a measure like nDCG, basically changing the DCG score of the ideal ranking for normalization. This filtering may have a more interesting impact on PGC, since it can substantially change the ideal rankings against which actual rankings are compared. For our experiments, we use public data linked from Sakai et al. [39, 40] or available through NTCIR[2].

Table 4.1 shows agreement with SERP preferences and sensitivity for selected measures from Sakai and Zeng [40]. Of the 35 measures they define and explore, the wpref6 measure provides the best agreement on the Derived 2019 preferences, while the wpref5 measure provides the best agreement on the Real preferences. PGC outperforms wpref6 on the Derived 2019 judgements, and would be tied for fifth (along with wpref2) on the Real judgments. nDCG outperforms all other measures on agreement, even ERR, which has outperformed nDCG on click metrics [12], but does poorly on this experiment, placing well below most preference metrics.

Sensitivity reflects the degree to which an evaluation measure can recognize significant differences between systems. The sensitivity values in Table 4.1 suggest the sensitivity may depend more on the judgment set than on the measure, where measures based on the filtered 2020 judgments are noticeably more sensitive than those based on the unfiltered 2019 judgments. PGC appears more generally sensitive than other measures, although the Combined preferences show that this again depends on the judgments. Nonetheless, from Table 4.1 we conclude that PGC appears to be an adequate substitute for both traditional and preference evaluation measures on both graded and explicit preference judgments, and even on combinations of judgments.

Figure 4.1 illustrates consistency for selected measures from Table 4.1. The graph in the upper left plots PGC based on real preferences (Real) with PGC based on preferences derived from graded judgments (Derived 2019). With Kendall's $\tau = 0.7524$, the correlation is weaker than that between nDCG and PGC based on preferences derived from Derived 2019 ($\tau = 0.9619$) and weaker than that between wpref5 and PGC based on Real preferences ($\tau = 0.8476$). The graph in the lower right plots nDCG vs. wpref5, completing the consistency comparison between these four measures.

[2] http://research.nii.ac.jp/ntcir/ntcir-9/index.html

Measure	Judgments	Sensitivity
Compatibility	Total	76.5%
PGC	**Real**	**73.7%**
PGC	**Combined**	**73.4%**
nDCG@3	Graded	71.7%
PGC	**Derived**	**71.3%**

Table 4.2: Sensitivity for PGC over the various judgment sets from Clarke et al. [15] along with nDCG@3 and compatibility measures from that paper.

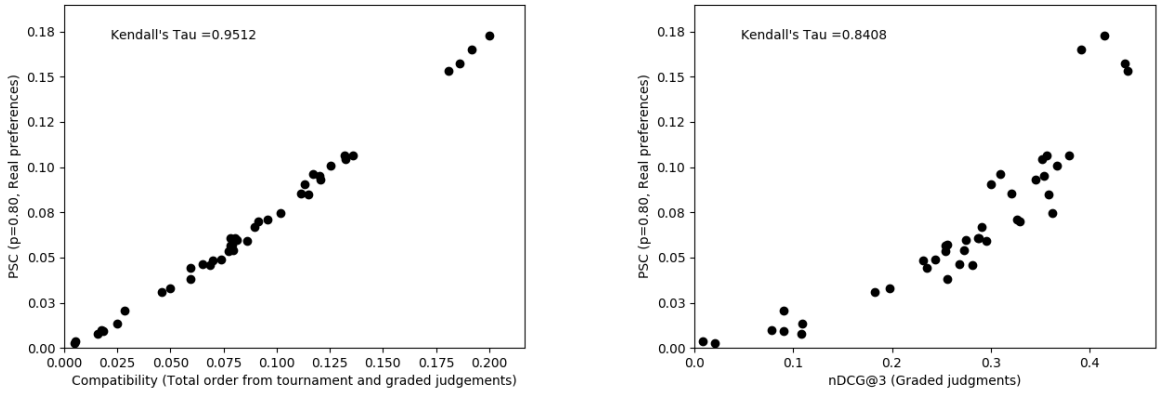


Figure 4.2: Scatter plots comparing measures from Clarke et al. [15] with PGC. Each point is an experimental run submitted to the TREC 2019 CAst Track. Measures are averaged over 173 questions. For PGC and compatibility, $p = 0.80$.

The stronger correlations in the lower left and upper right, suggest again that the order of rankers under these measures may depend more on properties of the judgments than on properties of the measures. PGC and nDCG are based on entirely different approaches to evaluation yet we see a strong correlation between them ($\tau = 0.9619$). In particular, PGC does not require us to interpret relevance grades as gain values. This strong correlation echoes the results of Clarke et al. [16] who also report a strong correlation between nDCG and the compatibility measure discussed in the next section. Although this measure requires a total order, and PGC works with any preference graphs, if a total order is treated as a preference graph, the measures are the same.

4.2 Clarke et al.

The results of the previous section suggest that the preference graph may be more important than the measure in the evaluation of search results. Clarke et al. [15] extend their prior work [14, 16] with this specific goal, focusing preference judgments on accurately identifying the top-k items, rather than finding a total order for an entire pool of n items. Starting with graded judgments, they structure explicit preference judgments as a tournament. Items losing too many match-ups are eliminated from further consideration until the top-k items are found. In their paper $k = 5$ and they do not explore other values of k. When comparing modern neural rankers on a question answering task, they demonstrate that these focused judgments can recognize significant improvements in quality missed by the traditional graded judgments.

As an evaluation measure they employ the compatibility framework introduced in their prior work [14, 16]. Since we also adopt this framework for this paper, the key ideas have already been introduced in Section 3.5. The key difference (and a central contribution of this paper) is that they require a total order, and their method cannot be used on an arbitrary collection of judgments. After running the tournament to identify the top-5 answers, they discard the judgments for evaluation purposes. Instead, the top-5 answers are added as five new relevance levels above the original values. While this approach is sufficient to demonstrate the value of focused top-k preferences, their measure cannot work with an arbitrary collection of preferences, limiting its generality.

The preference judgments of Clarke et al. [15] were created to augment graded judgments developed as part of the TREC 2019 CAsT question answering task [20]. For this task, questions were structured into information-seeking conversations, and participants were asked to provide answers at each turn of the conversation. Answers were drawn from a corpus of passages extracted from a variety of web sources. Similar to many TREC tracks, participating groups returned ranked lists of answers for each question, which were pooled down to depth 10 for judging. In total, 173 questions were judged on a typical five-point graded relevance scale from "fully meets" down to "fails to meet". As the primary evaluation measure they report nDCG@3, which Clarke et al. [15] adopt for their comparison.

To augment the graded relevance judgments from the original TREC task, Clarke et al. [15] ran their tournament using workers recruited through Mechanical Turk. In total, these workers made 15,349 explicit preference judgments to augment the original 29,350 graded judgments, just over 50% additional preference judgments. In contrast, Sakai and Zeng [40] report that their assessors made 119,646 preference judgments to augment 1,548

graded judgments, i.e., over 7,729% additional judgments. For our experiments, we use the preference judgments as linked from Clarke et al. [15] and the experimental runs as posted on the TREC site[3]. The data supplied by Clarke et al. [15] includes only 14,573 judgments, where the difference represents pilot tests and other judgments not contributing to the tournament.

As in Section 4.1, we wish to explore a variety of preference graphs. We employ three preference graphs:

- **Real** — Preference graphs from the 14,373 explicit preference judgments made by the Mechanical Turk workers. On average, they contain 83 edges per question.
- **Derived** — Preference graphs from the original 29,350 TREC 2019 CAsT graded judgments. On average, they contain 5,552 edges per question.
- **Combined** – The union of Real and Derived. On average, these preference graphs contain 5,637 edges per question.

We also report nDCG@3 using the 29,350 TREC 2019 CAsT graded judgments (**"Graded"**). We report the original compatibility measure of Clarke et al. [15] as "compatibility". It uses the total order from the tournament (**"Total"**), which includes the Graded judgments.

Table 4.2 shows sensitivity values for PGC using these preference graphs, along with nDCG@3 and compatibility measures from Clarke et al. [15]. The sensitivity for their compatibility measure, using their total order, remains higher than PGC for any combination of judgments. Similarly, nDCG@3 using the Graded judgments exhibits higher sensitivity than PGC using the preference graphs derived from the same source.

Nonetheless, despite the drop in sensitivity, the ordering of rankers remains consistent between PGC and compatibility. The scatter plot on the left of Figure 4.2 shows PGC using the Real preference graphs vs. compatibility using the Total order from the tournament, which includes the Graded judgments. The relationship appears essentially linear. The four experimental runs in the upper right are clearly separated from the other runs. These are the experimental runs that Clarke et al. [15] identify as indicating the impact of partial preferences, particularly since this a large and statistically significant difference. In addition to BERT-based re-ranking, these runs employ query expansion methods to improve answer retrieval. Under nDCG@3, these runs still represent the top 4, but nDCG is unable to recognize a statistically significant difference. The scatter plot on the right shows nDCG@3 on the x-axis and PGC on the y-axis. By projecting the points onto the

[3] http://trec.nist.gov

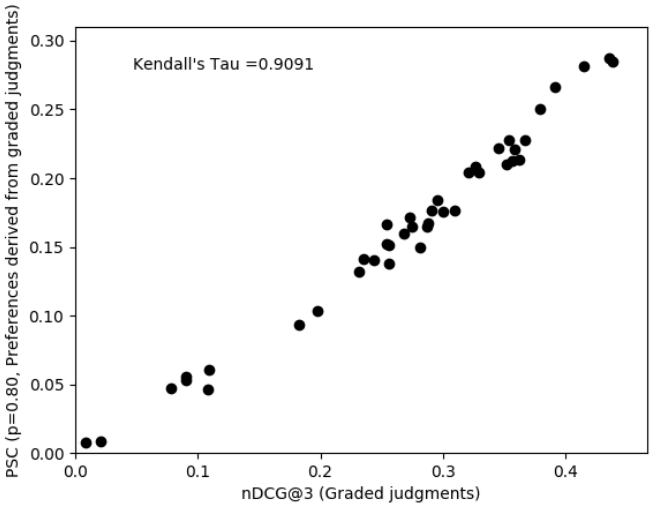


Figure 4.3: Scatter plot comparing nDCG@3 with graded judgments and PGC with preference graphs derived from those judgments. Each point is an experimental run submitted to the TREC 2019 CAsT Track. Measures are averaged over 173 questions. For PGC, $p = 0.80$.

x-axis, the lack of separation between the top-four runs and the remaining runs under nDCG@3 can be seen. This plot can be compared to (the nearly identical) Figure 9 in Clarke et al. [15].

Overall, these results support our observation that the order of rankers under these measures may depend more on properties of the judgments than on properties of the measures. As an additional example, Figure 4.3 plots PGC using the Derived judgments vs. nDCG@3 using the Graded judgments from which they're derived. Again, the relationship appears essentially linear, although the correlation coefficient is slightly lower ($\tau = 0.9091$). Although PGC and nDCG are based on entirely different approaches to evaluation, they produce similar outcomes on the same judgments.

Chapter 5

Image Dataset

Preference judgements and preference-based metrics for general web search have been researched for decades, while there is not much research about preference judgements for web image search. There are several differences between general web search results and web image search results: result placement, search intent and interaction mechanism[49]. The result of general web search is web pages, while the result of web image search is images. General web search displays the result as a sequential list, while web image search places the images as a grid-style. Users are able to browse more items in image results than general results without scrolling.

Due to these differences, traditional evaluation metrics for preference judgments may not be appropriate for evaluating web image search engines. Xie et al. [50] analyze user behaviors for a grid-based web image search result interface, and find that it is important and beneficial to take grid-based placement as consideration when evaluating image search results. Therefore, we incorporate the grid-based placement and Greedy PGC to compute the compatibility of image results. In this chapter, we experimentally compare Greedy PGC for image to recent effort to make preferences and preference judgments usable in practice.

5.1 Related Work

Previous work about web image search focuses on how diverse factors impact image search evaluation. Shao et al. [41] study the effects of context factor in image search and find that images with high variance obtain better correlation between users satisfaction and

evaluation measures. Tsukuda et al. [44] explore the influence of various search intents, from looking for a specific image to just browsing an image in an interesting field.

Most of the previous work is related to graded relevance evaluation measures. Zhang et al. [54] investigate the relationship between different measures and user satisfaction in image search, and find that offline measures such as NDCG, ERR and RBP are better correlated with user satisfaction than online measures such as dwell time and click rate. Xie et al. [50] propose three grid-based assumptions, and show that applying grid-based assumptions into evaluation metrics such as RBO and NDCG can achieve better performance in terms of correlation with user satisfaction.

To the best of our knowledge, Xie et al. [49] is the first attempt to study image search preference-based evaluation. They consider grid-based assumptions and implement a preference-based evaluation measure PWP for web image search. Another contribution they have is a publicly available dataset with preference judgments for over 40,000 web image pairs. In the following sections, we will apply our algorithm to this dataset, and compare this evaluation metric with PWP metric discussed in their paper.

5.2 Definitions

To evaluate the performance of a web image search engine and compute the compatibility, we need to define the actual ranking and ideal rankings. Unlike general web search results that are linear result lists ranking the most relevant item in the top, images search results display in a grid style, which cannot be simply ranked in a linear ranking list. We need to define users' examination sequence in such a grid-based placement firstly.

When users browse the image results, there are multiple patterns about the order of browsing images, such as middle position bias, and nearby principle [48]. Middle position bias means users follow a top to bottom sequence in the vertical direction, and focus more on the middle position in the horizontal direction. Nearby principle represents that users follow a top-bottom and left-right pattern and pay more attention to two images that are close to each other. In this paper, we do not discuss which pattern leads to a best compatibility. Instead, we compare the consistency of result of multiple examination sequences.

We will use one of examination sequences as an example, and explain how this examination sequence works in the grid images. The sequence of Euclidean distance assumes that users start from the top-left position, and follow a top to bottom and left to right sequence when they browse image results. It is assumed that the image in the top-left

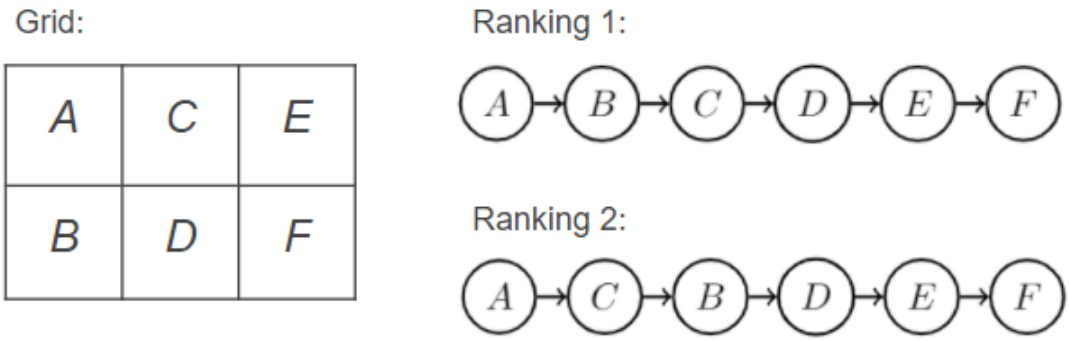


Figure 5.1: An image grid example.

position is always the first one to browse, and then by counting the distance between other positions and top-left position, users always look at the images with the least distance.

With such an examination sequence, we can transfer grid-based image results into a linear result list. Given a picture grid P with n row, and m columns, each item in the grid is labelled $(1,1),(1,2),\ldots,(1,m),(2,1),\ldots,(n,m)$, where (n,m) represents that the image placed at the nth row and mth column. The top-left position is denoted as $(1,1)$. Distance between $(1,1)$ and (n,m) is calculated as $D(n,m)=\sqrt{(n-1)^2+(m-1)^2}$. The rank for (n,m) is denoted as $R(n,m)$, where $R(1,1)=1$ represents the image placed at the top-left position is always ranked first and considered as the most relevant image. When deciding the next image in a ranking list from top to bottom, we always choose the item with the least distance to $(1,1)$.

As an illustrative example, consider transferring the grid in the left part of Figure 5.1 into a linear ranking list. Suppose we have 6 items in the grid, A,B,C,D,E,F, and this grid consists of 2 rows and 3 columns of items. In order to generate a linear ranking list, we

need to calculate the distance between the top-left item and the rest of items:

$$
\begin{aligned}
D(itemA) &= D(1,1) = 0 \\
D(itemB) &= D(2,1) = \sqrt{(2-1)^2 + (1-1)^2} = 1 \\
D(itemC) &= D(1,2) = \sqrt{(1-1)^2 + (2-1)^2} = 1 \\
D(itemD) &= D(2,2) = \sqrt{(2-1)^2 + (2-1)^2} = \sqrt{2} \\
D(itemE) &= D(1,3) = \sqrt{(1-1)^2 + (3-1)^2} = 2 \\
D(itemF) &= D(2,3) = \sqrt{(2-1)^2 + (3-1)^2} = \sqrt{5}
\end{aligned}
$$

We can find that D(2,1) is equal to D(1,2), then based on the ascending order of distances, there are two possible rankings: $R_1 = A, B, C, D, E, F$, and $R_2 = A, C, B, D, E, F$, shown in the right part of Figure 5.1.

5.3 Greedy PGC Algorithm for Image

From the above sections, we make two assumptions when evaluating the image search system: 1) images are placed as a grid style; 2) the sequence that users examine images is equal to the priority of image grid. In the following section, we will still employ examination sequence using Euclidean distance as an example of sequences, and apply this sequence into our algorithm. Euclidean distance is based on the distance between each image and the origin. The origin in an image grid is defined as the top-left position.

In order to adapt the image assumptions, we need to modify our Greedy PGC algorithm (discussed in the section 3.5). Instead of taking an actual ranking R as input, Greedy PGC for Image takes a preference graph G and an actual picture grid P as input, finds the ideal rankings I that are closest to the input grid P, and outputs the compatibility of the actual image grid P and the ideal ranking I. The pseudocode for Greedy PGC for Image is presented in Algorithm 5.

The overall structure of Greedy PGC for Image remains the same, while the way to determine the next node u is different due to the image grid input. In order to incorporate the image grid, the function RankSink and RankSource are modified into RankImageSink and RankImageSource.

Function RankImageSink takes a sink list N and an actual picture grid P, instead of actual ranking R. If there are any sinks that do not exist in the grid P, the function returns an arbitrary node n in such sinks; if all the sinks are in the image grid P, then

Algorithm 5 GreedyPGC for Image

Input
    Directed graph $G = (V, E)$, actual image/picture grid P
Output
    Preference Graph Compatibility
$s_1 \leftarrow \emptyset$, $s_2 \leftarrow \emptyset$
while $G \neq \emptyset$ **do**
    while G contains a sink **do**
        $u \leftarrow$ RankImageSink($\forall$ *sinks* $\in G, P$)
        $s_2 \leftarrow us_2$
        $G \leftarrow G \setminus$u
    while G contains a source **do**
        $u \leftarrow$ RankImageSource($\forall$ *sources* $\in G, P$)
        $s_1 \leftarrow s_1 u$
        $G \leftarrow G \setminus$u
    $u \leftarrow$ RankImageSource($\forall\ v \in V$ with maximum $\delta(v), P$)
    $s_1 \leftarrow s_1 u$
    $G \leftarrow G \setminus$u
$I \leftarrow s_1 + s_2$
return RBO(I, P)

Figure 5.2: Greedy PGC algorithm for image.

Algorithm 6 RankImageSink

Input
Node list N, actual image/picture grid P

Output
Node n with the longest distance from the origin

$p \leftarrow \emptyset$
for all node $n_i \in N$ **do**
  if n_i exists in P **then**
    $d \leftarrow$ distance between n_i and the origin in P
  else
    $d \leftarrow \infty$
  $p \leftarrow p + (n_i, d)$
return n with $max(d)$ in p

Algorithm 7 RankImageSource

Input
Node list N, actual image/picture grid P

Output
Node n with the shortest distance from the origin

$p \leftarrow \emptyset$
for all node $n_i \in N$ **do**
  if n_i exists in P **then**
    $d \leftarrow$ distance between n_i and the origin in P
  else
    $d \leftarrow \infty$
  $p \leftarrow p + (n_i, d)$
return n with $min(d)$ in p

Figure 5.3: Functions RankImageSink and RankImageSource in Greedy PGC algorithm for image.

the function returns the sink farthest away from the origin, where the origin refers to the top-left position. If there are multiple nodes with the same maximal distance, a random one will be picked. In line 8 of algorithm 3, instead of finding the rank of n_i in R, we change into finding the distance between n_i and the origin in P. The pseudocode for RankImageSink is presented in Algorithm 6.

Function RankImageSource takes a node list N and an actual picture grid P and chooses the vertex u nearest to the origin or an arbitrary vertex u that is not in the grid P. If multiple nodes have the same minimal distance from the origin, we pick one randomly. In the original function RankSource, we modify line 8 to find the distance from the origin to the current node n_i. The pseudocode for RankImageSource is presented in Algorithm 7.

Once the ideal ranking I is generated, we use RBO to calculate the similarity between actual image grid P and ideal ranking I, denoted as $RBO(I, P)$. RBO is to evaluate the similarity between two linear lists, while we take a grid and a linear list as inputs of RBO. In order to compute the RBO value, we need to transfer the actual image grid P into a linear actual ranking list R based on the ideal ranking I. We use Euclidean distance as an example of examination pattern and follow the steps in the section 5.2 to generate a linear ranking from an image grid. Based on the distance between each image's position and the origin, we always choose the image with the least distance as the next item in the ranking list from top to bottom. However, when there are multiple images with the same least distance, instead of having multiple possible result ranked lists, we choose the image ranked higher in the ideal ranking I as the next image appended to the result list R, which adapts our Greedy PGC for Image function.

As an illustrative example, consider the implementation of Greedy PGC on the preference graph in Figure 3.2 and the actual image grid in the Figure 5.4. Suppose we have 7 images in the preference graph G, A,B,C,D,F,G,H, and their actual rankings are placed as a grid P, where x, y and z in the grid represent the images that appear in the actual image grids but not in the preference graph. Greedy PGC for image executes on this graph and actual ranking grid as following steps:

1. Initially, there are 3 sinks in this graph, vertex D, F, and G. In the first inner loop of the algorithm, distance D(vertex F) > D(vertex D), D(vertex G) = inf, because vertex G is not in P. the order of prepending to s_2 is vertex G, F, D. Current s_2 = [D, F, G].

2. After removing vertex D, F, and G from the graph G, two sources, vertex A and H, are left in the graph G. In the second inner loop of the algorithm, based on

the grid P, distance D(vertex A) < D(vertex H), where D(vertex H) = inf. Node with shortest distance is chosen first, so the order of appending to s_1 is vertex A, H. Current s_1 = [A, H].

3. Vertex B and C are left in the current graph G, where $\delta(A) = \delta(B)$. According to the grid P, vertex C is the origin, so distance D(vertex C) < D(vertex B). Vertex C is chosen to be appended to s_1. Current s_1 = [A, H, C].

4. In the current graph G, vertex B is left. In the second iteration of the outer loop, vertex B is removed from graph G and prepended to s_2. Current s_2 = [B, D, F, G].

5. Graph G becomes an empty graph, and the iteration stops. Ideal ranking $I = s_1 + s_2$ = [A, H, C, B, D, F, G], which is represented in the right-upper part of Figure 5.4.

6. To calculate RBO between ideal ranking I and actual image grid P, we transfer actual image grid P into a linear ranking list R. By calculating the distance between each image position and the origin, D(vertex C) > D(vertex A) = D(vertexB) > D(vertex x) > D(vertex y) > D(vertex D) > D(vertex F) > D(vertex z). For vertex A and B, they have the same distance while vertex A is ranked higher than vertex B in the ideal ranking I. The linear actual ranking R = [C, A, B, x, y, D, F, z], and it is shown as the right-bottom part of Figure 5.4.

7. RBO between ideal ranking I and actual ranking R is calculated as following, where $p = 0.95, DEPTH = 7$:

$$\begin{aligned} RBO(I, R) &= (1 - 0.95) \cdot 3.249 \\ &= 0.162 \end{aligned}$$

   The function Greedy PGC for Image returns the value of RBO.

5.4 Experiments

5.4.1 Data

Xie et al. base their work on a data collection of web search images. They randomly sampled 150 torso queries that had a fair amount of search volume and engagement in October 2017 from the Sogou image search engine. Based on these queries, they used two

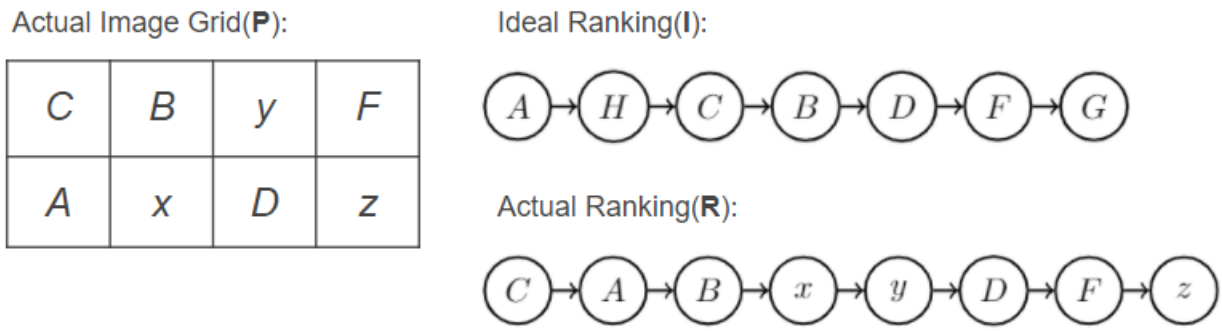


Figure 5.4: An example of input and output of Greedy PGC for Image.

popular Chinese search engines, Baidu and Sogou, to collect all the images shown in the web pages without scrolling under the maximum resolution settings of the web browser, which were the top three rows of image results. Discarding the pornographic queries and advertising image results, they collected 102 queries and 2,919 images in total, and the number of images for each query and each search engine was ranging from 8 to 22.

To study image preferences, they paired all images with all other images for each query and recruited three assessors to provide preference judgments for 41,538 image-image pairs, which they called image pairs annotation. Five levels of preference relevance were used, "definitely left", "left", "tie", "right", "definitely right". Figure 5.5 shows an example of a preference judgment interface of two image results for the query "Fortifications of Xi'an" from [49].

They employed three assessors, different from previous assessors, to annotate 100-point relevance scores for 2,919 images, which they call relevance data.

They hired 5 professional assessors to provide weak preference judgments for 102 Baidu-Sogou pairs. Three levels of preference relevance were adopted, "Sogou wins", "tie", "Baidu wins". 29 queries were judged as Sogou wins, 46 queries were judged as two systems tie, 27 queries were judged as Baidu wins. This dataset was called SERP-level preferences, which was considered as a golden standard to evaluate their evaluation metrics.

The statistics of datasets are shown in table 5.1.

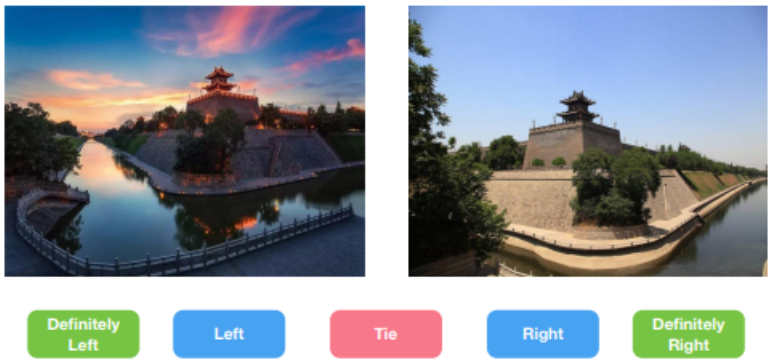


Figure 5.5: An example of preference judgment interface from Xie et al. [49]. The given two images are results of the query "Fortification of Xi'an".

Dataset	#Queries	#Images	#Image pairs
Image pairs annotation	102	2,919	41,538
Relevance data	102	2,919	2,919
SERP-level preferences	102	/	102

Table 5.1: Statistics of datasets in Xie et al. experiments.

5.4.2 Measures of Xie et al.

Xie and his colleagues develop a preference-based evaluation metric for web image search engines, called Preference-Winning-Penalty (PWP). PWP takes preference judgments for image pairs from two search systems, uses major voting to decide the winner of pairs, and outputs the probability that one system is preferred over the another. PWP consists of three variables, preference matching rate (PMR), winning rate (WR) and penalty for bad cases (PB). PMR measures image preferences of one system, while WR and PB focus on preference judgments for two image systems at the same time.

Preference matching rate (PMR) is the ratio of correctly ordered pairs based on users' preference judgments to the total number of items in the set, where the word "ordered" in the correctly ordered pairs refers to the orders that transfer a grid-based image result into a linear list. Xie and his colleagues applied four examination orderings into PMR in their experiments:

1. Default(PMR_D): Users follow the pattern of left to right, top to bottom.

2. Weighted(PMR_W): PMR_W is a weighted version of PMR_D. For image results at ranks i and j such that $j > i$, the weight w_{ij} is defined as

$$w_{ij} = \frac{1}{\log_2(j+1)}.$$

   PMR_W is the sum of all weights w_{ij} over the total number of pairs i, j such that the item ranked at i is preferred to the item ranked at j and the ranks $i < j$.

3. Middle position bias(PMR_M): Vertically, users follow a top to bottom pattern, while users focus more on the middle positions horizontally.

4. Nearby principle(PMR_N): PMR_N combines the examination sequence assumption behind PMR_D and nearby principle assumption together. Nearby principle assumes users always focus on the closest distance between two image positions, then the distance function D is defined as

$$D = max(|r_i - r_j|, |c_i - c_j|),$$

   where r represents the row number and c represents the column number of image results.

Given two different search systems S_1 and S_2, winning rate (WR) for system S_1 is the ratio of the number of pairs that the image from system S_1 is preferred to the total number of pairs between system S_1 and S_2. WR evaluates the performance of a search system by the number of good-quality images compared with another search system.

Penalty for bad cases (PB) takes bad cases in the preference judgments into consideration for two search systems. The "bad case" is defined as give two search systems S_1 and S_2, an image result i in the pairs from S_1 is a bad case if, and only if, j is preferred than i for any image result j in the pairs from S_2. Given two search systems, PB for system S_1 is calculated by using the number of bad cases in system S_1 as the exponent of γ, where $0 < \gamma < 1$.

With PMR, WR and PB, preference-winning-penalty (PWP) evaluates a search system S_1 by considering preference judgments from two search systems S_1 and S_2. It generates a preference score for the system S_1:

$$PWP(S_1|S_2) = (\lambda PMR(S_1) + (1-\lambda)WR(S_1|S_2)\dot{P}B(S_1|S_2))),$$

where λ is a trade-off parameter to balance the PMR and WR of system S_1 given system S_2.

5.4.3 Results of Xie et al.

Table 5.6 shows agreements between SERP level preferences and users' search engine preferences for measures from Xie and his colleagues[49] ($\gamma = 0.1, \lambda = 0.7$). Of the four measures (PWP, PB, PMR_N, and WR), PWP measure obtains the best agreement It has the most cases of: for each query, when PWP score for Sogou is higher, SERP level preference also judges Sogou is preferred; when PWP score for Baidu is higher, SERP level preference thinks Baidu is preferred. PB, WR and PWP are consistent in terms of agreements for each dataset, which achieve more agreements when both of these measures and SERP level preferences prefer images from Sogou than Baidu.

The results of four variables are different in terms of Sogou's preferences within SERP level preferences. Given that Sogou is preferred in SERP level preferences, PWP and PMR_N give dominant preferences on Sogou, while PB has more tie situations and WR provides significantly more numbers of Sogou preferences. On the other hand, from the perspective of Baidu's preferences in SERP level preference, the results of four variables are consistent in presenting noticeably more cases that Baidu is preferred. For ties in SERP level preference, PWP, WR and PB suggest that Baidu is better, while PMR_N is

	SERP level preference			
WR		Sogou better	Baidu better	Tie
	Sogou better	9	3	3
	Baidu better	19	25	42
	Tie	0	0	1

(a) WR, $P(\chi^2 > 3.8182) = 0.0507$, $p = 0.000017$ (one-tailed)

	SERP level preference			
PB		Sogou better	Baidu better	Tie
	Sogou better	8	0	4
	Baidu better	3	15	9
	Tie	17	13	33

(b) PB, $P(\chi^2 > 15.7576) = 0.00007$, $p = 0.038778$ (one-tailed)

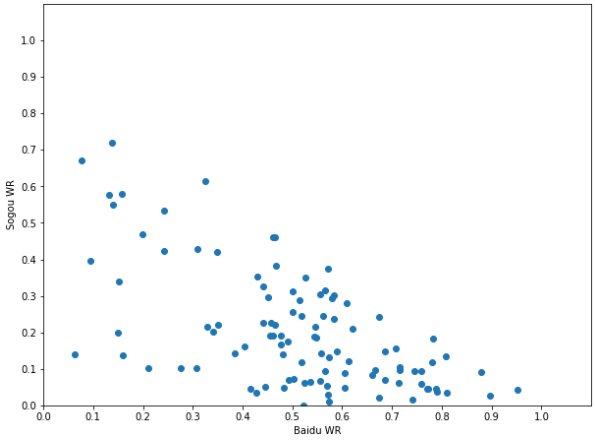
	SERP level preference			
PMR_N		Sogou better	Baidu better	Tie
	Sogou better	18	12	25
	Baidu better	10	16	21

(c) PMR_N, $P(\chi^2 > 2.5846) = 0.1079$, $p = 0.34425$ (one-tailed)

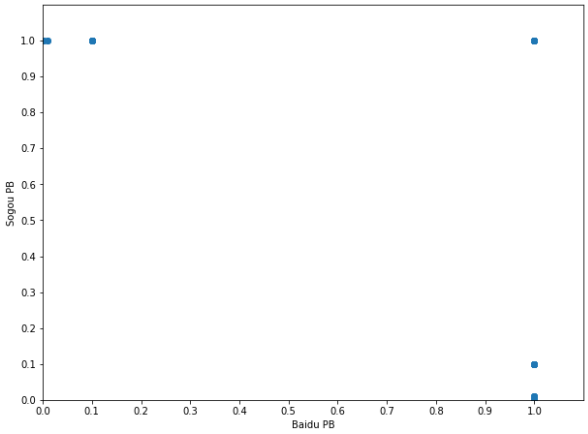
	SERP level preference			
PWP		Sogou better	Baidu better	Tie
	Sogou better	17	3	10
	Baidu better	11	25	36

(d) PWP, $P(\chi^2 > 15.2444) = 0.00009$, $p = 0.02251$ (one-tailed)

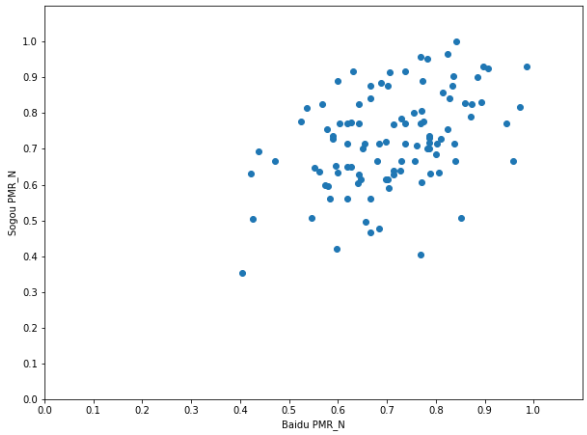
Figure 5.6: Agreements between SERP level preferences and users' search engine preferences for PMR_N, WR, PB, PWP from Xie et al.[49].



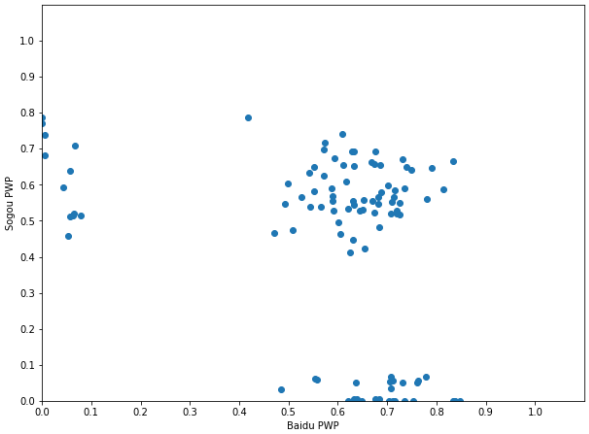
(a) WR.
Kendall's τ = -0.4052



(b) PB.
Kendall's τ = -0.2025



(c) PMR_N.
Kendall's τ = 0.2882



(d) PWP.
Kendall's τ = -0.2293

Figure 5.7: Scatter plots comparing Sogou scores and Baidu scores for PMR_N, WR, PB, PWP from Xie et al.[49].

slightly reversed. Unlike other measures, PB exists tie situations and the number of ties is indispensable.

Compared the total number of preferable Sogou images to the number of preferable Baidu images given by each measure, WR, PB and PWP suggest that there are more and better images from Baidu than those from Sogou, while PMR_N judges Sogou images are better in most of time.

A chi-squared test of independence is applied to examine the relationship between SERP level preferences and users' preferences generated by different measures from Xie et al [49]. The null hypothesis is that SERP level preferences are independent of users' preferences evaluated by multiple measures. We exclude the tie situations from the tables, and calculate only the comparison between Sogou better and Baidu better. In this case, the degree of freedom in the chi-squared test is 1.

For PMR_N and WR measures, since the p-value is greater than the chosen significance level ($p = 0.05$), we do not reject the null hypothesis, which means that there is no association between SERP level preferences and PMR_N preferences ($\chi^2(1, N = 102) = 2.58462, p < 0.1079$), and that between SERP level preferences and WR ($\chi^2(1, N = 102) = 3.8182, p < 0.0507$). The relation between SERP level preferences and PB measure ($\chi^2(1, N = 102) = 15.7576, p < 0.00007$), and that between SERP level preferences and PWP measure ($\chi^2(1, N = 102) = 15.2444, p < 0.00009$) are all statistically significant.

Binomial test for one tail is applied to investigate if images from Sogou are likely to be equally good to images from Baidu. The null hypothesis is that users' preferences generated by different measures from Xie et al. [49] are equally likely to be Sogou images and Baidu images. We exclude the tie situations from the tables, and calculate only the comparison between Sogou better and Baidu better. If we have a significance level of 5%, then the result of PMR_N measure ($0.34425 > 5\%$) indicates that we cannot reject the null hypothesis that Sogou images and Baidu images are equally good. On the other hand, the binomial test results of WR ($0.000017 < 5\%$), PB ($0.038778 < 5\%$), and PWP ($0.02251 < 5\%$) significantly reject the null hypothesis, which means that for these measures, users do not prefer two image search engines equally but prefer Baidu over Sogou.

From table 5.6, we can conclude that PWP outperforms all other measures on agreements, and Baidu appears to be a better image search engine, while four measures are not consistent with each other in multiple perspectives.

Figure 5.7 illustrates the comparison between scores for Baidu dataset and Sogou dataset from four metrics. PMR_N graph has similar minimal values with PWP graph, which are all ranging from 0.3 for Sogou dataset and 0.4 for Baidu dataset, while the maximal scores for both datasets in PMR_N graph are higher than the ones in PWP graph. PB

graph is a special plot, which contains only 1, 0.1 and numbers that are extremely close to 0. This can be explained by the equation of PB, which is the exponent of λ ($0 < \lambda < 1$). PWP graph consists of three clusters. The plot shape in the right middle part of the graph is similar to the plot shape in PMR_N graph, while the scatter points in the rest parts are pretty extreme. For those points, when one system achieves a score larger than 0.4, another system must obtain a particularly low score, ranging from 0.0 to 0.1.

With Kendall's $\tau = -0.4052$, the anti-correlation between Sogou WR score and Baidu WR score is strongest than that between Sogou PWP score and Baidu PWP score ($\tau = -0.2293$), and stronger than that between Sogou PB score and Baidu PB score ($\tau = -0.2025$).

From figure 5.7, we can only conclude that PMR_N is somehow correlated with part of PWP, but it is not clear to conclude that there is a strong correlation between four different metrics.

5.4.4 Results of Greedy PGC for Image

We wish to explore preference graphs with multiple examination orderings. We employ six examination orderings, where the first three of them follow the examination sequence setting of Xie and his colleague:

1. Default: Users follow the pattern of left to right, top to bottom.
2. Weighted default: This is a weighted version of default ordering. For image results at ranks i and j such that $j > i$, the weight w_{ij} is defined as
$$w_{ij} = \frac{1}{\log_2(j+1)}.$$
3. Bottom-top, right-left: Users follow the reversed version of default ordering, which is bottom to top, right to left.
4. Manhattan distance: This combines the examination sequence assumption behind default examination sequence and Manhattan distance together. It assumes users always focus on the closest Manhattan distance between two image positions, then the distance function D is defined as
$$D = |r_i - r_j| + |c_i - c_j|,$$
where r represents the row number and c represents the column number of image results.

Left-right, top-bottom				
	SERP level preference			
Greedy PGC		Sogou better	Baidu better	Tie
	Sogou better	6	3	4
	Baidu better	22	25	42

(a) Left-right and top-bottom.
$P(\chi^2 > 1.19149) = 0.275$,
$p = 0.000001$ (one-tailed)

Weighted Default				
	SERP level preference			
Greedy PGC		Sogou better	Baidu better	Tie
	Sogou better	8	6	6
	Baidu better	20	22	40

(b) Weighted default.
$P(\chi^2 > 0.380952) = 0.5371$,
$p = 0.000154$ (one-tailed)

Middle Position Bias				
	SERP level preference			
Greedy PGC		Sogou better	Baidu better	Tie
	Sogou better	6	3	4
	Baidu better	22	25	42

(c) Middle position bias.
$P(\chi^2 > 1.19149) = 0.275$,
$p = 0.000001$ (one-tailed)

Bottom-top, right-left				
	SERP level preference			
Greedy PGC		Sogou better	Baidu better	Tie
	Sogou better	8	6	6
	Baidu better	20	22	40

(d) Bottom-top and right-left.
$P(\chi^2 > 0.380952) = 0.5371$,
$p = 0.000154$ (one-tailed)

Figure 5.8: Agreements between SERP level preferences and Greedy PGC preferences with six examination sequences

Manhattan Distance				
	SERP level preference			
Greedy PGC		Sogou better	Baidu better	Tie
	Sogou better	7	4	4
	Baidu better	21	24	42

(e) Manhattan distance.
$P(\chi^2 > 1.01818) = 0.313$,
$p = 0.000005$ (one-tailed)

Euclidean Distance				
	SERP level preference			
Greedy PGC		Sogou better	Baidu better	Tie
	Sogou better	8	4	5
	Baidu better	20	24	41

(f) Euclidean distance.
$P(\chi^2 > 1.69697) = 0.1927$,
$p = 0.000017$ (one-tailed)

Figure 5.8: Agreements between SERP level preferences and Greedy PGC preferences with six examination sequences

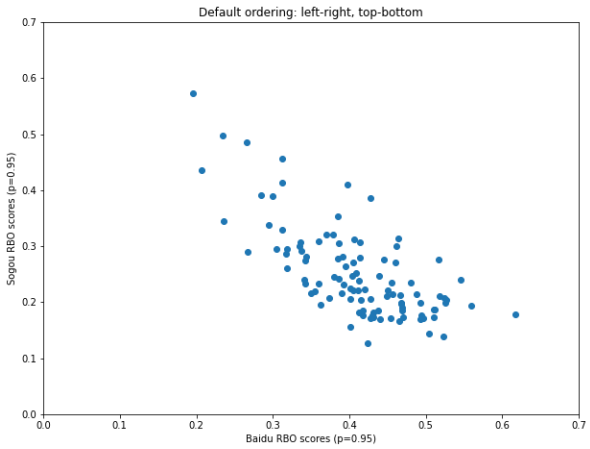
5. Euclidean distance: This combines the examination sequence assumption behind default examination sequence and Euclidean distance together. It assumes users always focus on the closest Euclidean distance between two image positions, then the distance function D is defined as

$$D = \sqrt{(r_i - r_j)^2 + (c_i - c_j)^2},$$

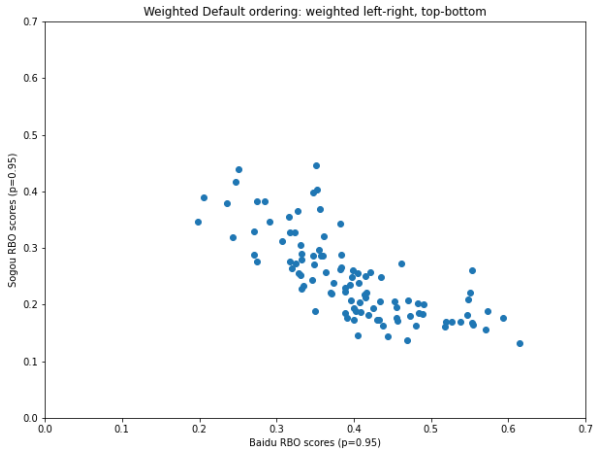
where r represents the row number and c represents the column number of image results.

Table 5.8 represents the agreement between SERP level preferences and Greedy PGC for images with six different orderings. There is no significant difference between these tables from the perspective of agreements. The number of agreements for all measures are ranging from 30 to 32, and Euclidean distance measure has a slightly better number of agreements. The number of cases that both SERP preferences and Greedy PGC prefer Sogou is considerably less than those cases that both prefer Baidu.

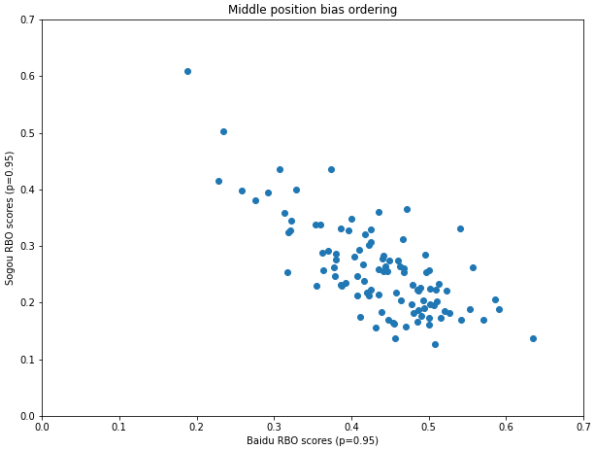
The results for six orderings are also consistent in terms of the ratio of Sogou preference and Baidu preference judged by Greedy PGC. The number of cases that Greedy PGC prefers Baidu for all examination sequences is substantially more than the number of cases that Greedy PGC prefers Sogou.



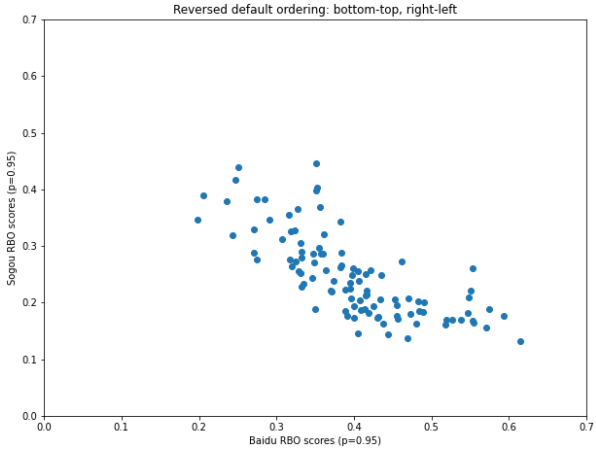
(a) Left-right and top-bottom.
Kendall's τ = -0.4906



(b) Weighted default.
Kendall's τ = -0.6063

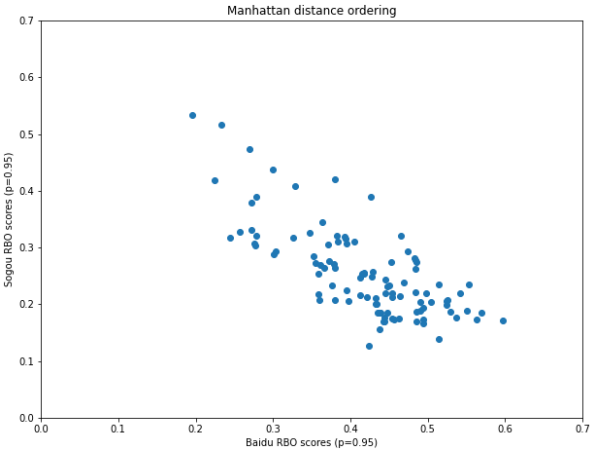


(c) Middle position bias.
Kendall's τ = -0.4932

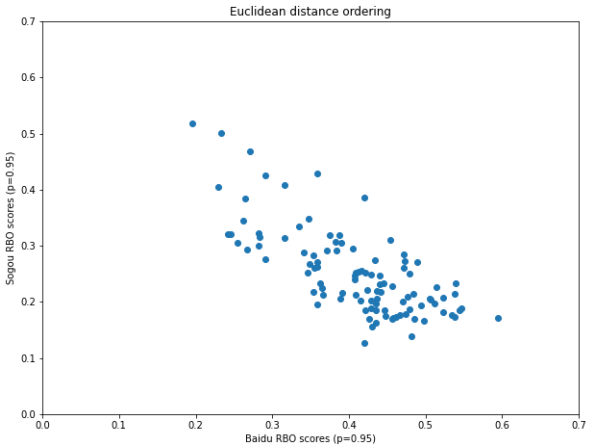


(d) Bottom-top and right-left.
Kendall's τ = -0.6056

Figure 5.9: Scatter plots comparing Sogou scores and Baidu scores generated by Greedy PGC with six examination sequences



(e) Manhattan distance.
Kendall's τ = -0.5058



(f) Euclidean distance.
Kendall's τ = -0.5134

Figure 5.9: Scatter plots comparing Sogou scores and Baidu scores generated by Greedy PGC with six examination sequences

A chi-squared test of independence is applied to examine the relationship between SERP level preferences and users' preferences generated by Greedy PGC in different examination orderings. The null hypothesis is that SERP level preferences are independent of users' preferences evaluated by Greedy PGC. We exclude the tie situations from the tables, and calculate only the comparison between Sogou better and Baidu better. In this case, the degree of freedom in the chi-squared test is 1.

For Greedy PGC with all examination orderings we have tested, the p-values are greater than the chosen significance level ($p = 0.05$), and we cannot reject the null hypothesis. The relations between SERP level preferences and these ordering measures are not significant: $\chi^2(1, N = 102) = 1.19149, p < 0.275$ for left-right and top-bottom and middle position bias orderings, $\chi^2(1, N = 102) = 0.380952, p < 0.5371$ for weighted default and bottom-top and right-left, $\chi^2(1, N = 102) = 1.01818, p < 0.313$ for Manhattan distance and $\chi^2(1, N = 102) = 1.69697, p < 0.1927$ for Euclidean distance.

Binomial test for one side is applied to investigate if images from Sogou are likely to be equally good to images from Baidu. The null hypothesis is that users' preferences evaluated by Greedy PGC are equally likely to be Sogou images and Baidu images. We exclude the tie situations from the tables, and calculate only the comparison between Sogou better and Baidu better. A binomial test indicated that the proportion of Baidu preference of 0.84 ($p = 0.000001$) for left-right and top-bottom and middle position bias orderings, of 0.75

($p = 0.000154$) for weighted default and bottom-top and right-left, of 0.8 ($p = 0.000005$) for Manhattan distance ordering, and of 0.79 ($p = 0.000017$) for Euclidean distance ordering were higher than the expected 0.5. These results represent that the null hypothesis is significantly rejected and Greedy PGC prefers Baidu images over Sogou images.

From table 5.8, it is evident to conclude that in terms of agreements, Greedy PGC using Euclidean distance somewhat outperforms all other measures although the results of all measures do not have much difference, and Greedy PGC suggests images from Baidu dataset is preferable.

Figure 5.9 illustrates the consistency for Greedy PGC for images with different examination sequences. All the plots have similar shape, and represent that there exists an anti-correlation between Sogou RBO scores and Baidu RBO scores for all the orderings. Weighted default measure ($\tau = -0.6063$) has a stronger anti-correlation than bottom-top, right-left measure ($\tau = -0.6056$), than Euclidean distance measure ($\tau = -0.5134$), than Manhattan distance measure ($\tau = -0.5058$), than middle position bias ($\tau = -0.4932$), than default measure ($\tau = -0.4906$). In terms of minimal and maximal values, Baidu RBO scores range from 0.2 to 0.7, while Sogou RBO scores range from 0.1 to 0.6.

From figure 5.9, it is observable that RBO scores are consistent in different order settings for each search engine, and the range of Baidu RBO scores is higher than the range of Sogou RBO scores. It is also noticeable that there is a strong anti-correlation between Sogou RBO scores and Baidu RBO scores in weighted default ordering.

5.4.5 Graded Relevance Results

Two ways are applied to calculated NDCG@10 based on the graded relevance judgments. One of NDCG@10, which we call "original NDCG@10", follows the NDCG@10 measure in the work of Xie et al.[49]. For each search engine, both actual ranking and ideal ranking consists only of the images from this system. Actual ranking is sorted by default examination sequence and ideal ranking is sorted by relevance grades. Another NDCG@10, which we call "combined NDCG@10", is similar to the previous NDCG@10 measure, except the ideal ranking. Unlike the previous NDCG@10 measure, the ideal ranking in this measure consists of merged images from both search engines. In this case, ideal rankings are the same when calculating NDCG@10 of different systems.

Figure 5.10 represents the agreement between SERP level preferences and NDCG@10 preferences based on the graded relevance judgments.

A chi-squared test is calculated to examine the relationship between SERP level preferences and NDCG@10 preferences. The test shows that the relation for both NDCG@10

	SERP level preference			
NDCG @10		Sogou better	Baidu better	Tie
	Sogou better	16	7	16
	Baidu better	12	21	30

(a) Original NDCG@10.
$P(\chi^2 > 5.9763) = 0.0145$
$p = 0.114551$ (one-tailed)

	SERP level preference			
NDCG @10		Sogou better	Baidu better	Tie
	Sogou better	11	3	2
	Baidu better	17	25	44

(b) Combined NDCG@10.
$P(\chi^2 > 6.09524) = 0.01355$
$p = 0.000154$(one-tailed)

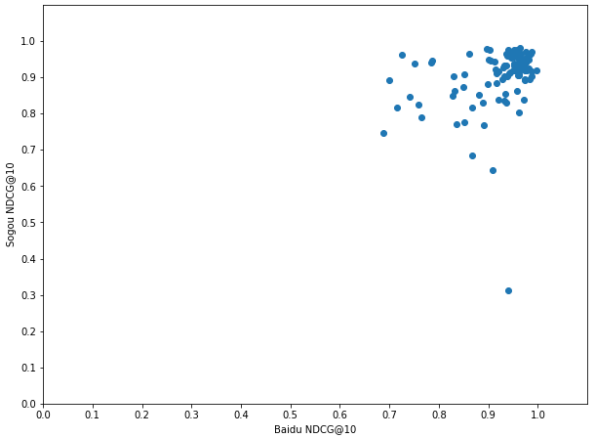
Figure 5.10: Agreements between SERP level preferences and NDCG@10 with relevance grades.

are statistically significant: $\chi^2(1, N = 102) = 5.9763, p < 0.0145$ for combined NDCG@10; and $\chi^2(1, N = 102) = 6.09524, p < 0.01355$ for combined NDCG@10.

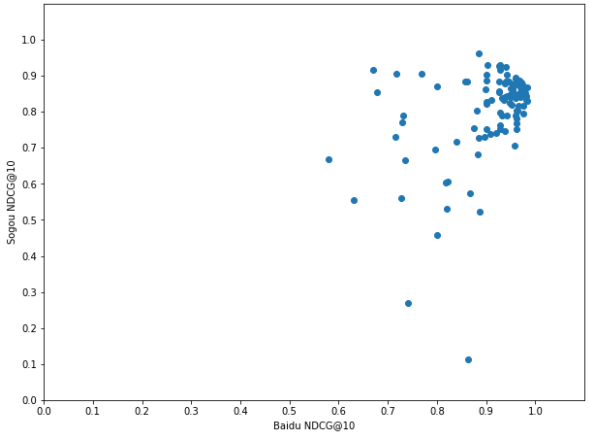
A binomial test is used to investigate whether images from Sogou are likely to be equally good to images from Baidu. The result of the test indicates that for original NDCG@10, we cannot reject the null hypothesis and Sogou and Baidu are equally good in terms of image preferences. For combined NDCG@10, it is significant to reject the null hypothesis and Baidu is preferred over Sogou.

On the other hand, both NDCG@10 prefers Baidu images for dominantly more queries than Sogou images, regardless of SERP level preferences.

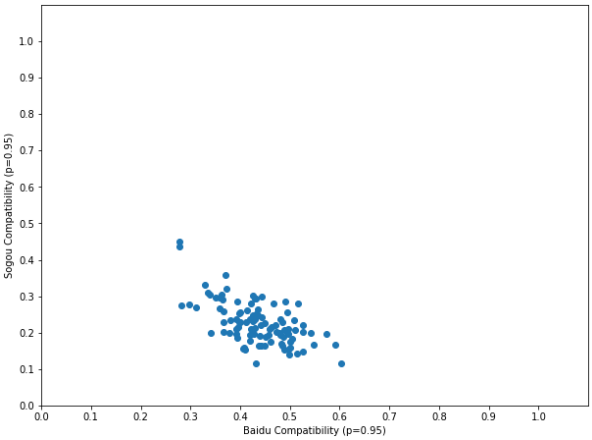
Figure 5.11 demonstrates the comparison between Baidu score and Sogou score based on graded relevance judgments. Figure 5.11a shows the scatter plot about original NDCG@10 with Kendall's $\tau = 0.2475$. Baidu scores and Sogou scores are mainly ranging from 0.6 to 1.0. Figure 5.11b plots Baidu scores and Sogou scores generated by combined NDCG@10, where ideal ranking is based on relevance grades and actual ranking follows the default examination sequence (left-right, top-bottom). Baidu scores range from 0.5 to 1, while Sogou scores range from 0.1 to 1. With Kendall's $\tau = 0.2079$, Baidu NDCG@10 score is somehow correlated with Sogou NDCG@10. Figure 5.11c is the scatter plot for Baidu compatibility and Sogou compatibility, which uses another method (RBO) to evaluate Baidu and Sogou search engines. Baidu scores range from 0.2 to 0.7, while Sogou scores range from 0.0 to 0.5. Baidu compatibility is anti-correlated with Sogou compatibility



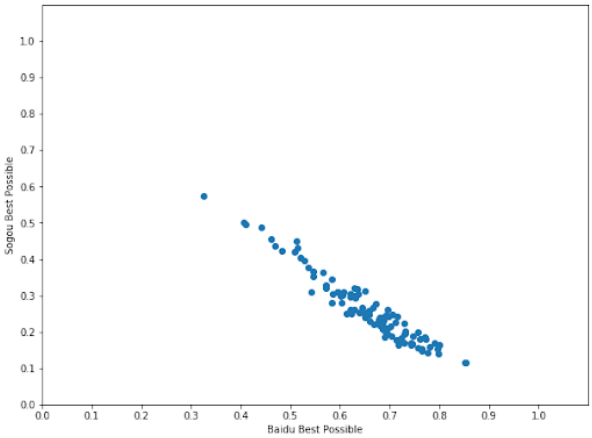
(a) Scatter plot for original NDCG@10.
Kendall's τ = 0.2475



(b) Scatter plot for combined NDCG@10.
Kendall's τ = 0.2079



(c) Scatter plot for NDCG-based compatibility.
Kendall's τ = -0.3791



(d) Scatter plot of best possible result from each engine.
Kendall's τ = -0.8195

Figure 5.11: Scatter plots comparing Sogou scores and Baidu scores generated by NDCG and compatibility with relevance grades.

(Kendall's τ = -0.3791). Figure 5.11d shows the relationship between Baidu best possible and Sogou best possible, which calculates the similarity between the ideal ranking based on relevance grades and best possible ranking from the ideal ranking for each search engine. Baidu best possible and Sogou best possible are strongly anti-correlated with Kendall's τ = -0.8195. Baidu scores range from 0.3 to 0.9 while Sogou scores range from 0.1 to 0.6.

From figure 5.11, we can conclude that based on the result of graded judgments, Baidu is considered as a better image search engine than Sogou by different evaluation measures.

Overall, most results support that Baidu is a better image search engine than Sogou, even though SERP level preferences suggest that Sougou and Baidu are equally good. We contacted Xie et al. [49] and they confirmed that we have reported their numbers correctly. Of the measures of Xie et al. [49], winning rate (WR) appears to give results that are closest to Greedy PGC, including the ratio of agreements and the anti-correlation between two search engines. WR directly compares between two search engines Baidu and Sogou, while PMR compares search engines themselves and does not consider the difference between them. WR is a small proportion of PWP while PMR is weighted heavily in PWP measure.

Chapter 6

Conclusion

This thesis proposes a preference-based evaluation measure called Greedy PGC that takes a collection of preferences as a directed multigraph, computes the ideal ranking by finding a minimal feedback arc set, and calculates the maximum similarity between actual ranking and ideal ranking as output.

For the evaluation of general web search results, we compare Greedy PGC and evaluation measures from recent work. Our results show that Greedy PGC matches or exceeds the performance of evaluation measures proposed in recent research.

For the evaluation of image web search results, we adapt Greedy PGC to suit image grid assumptions, and apply six examination sequences into our algorithm. We use functions RankSink and RankSource that choose the document based on the ranking with the functions that choose the image based on the distance of different examination sequences. Our results prove the consistency of Greedy PGC on different examination sequences and traditional evaluation measures.

There are several advantages of Greedy PGC over other evaluation methods. Firstly, Greedy PGC is not restricted to handle document results, but also able to evaluate web image search results. Secondly, collections of preferences could be any arbitrary collections. Conflicts, redundancies and incompleteness are acceptable and allowable in a data collection. Thirdly, Greedy PGC is able to take any preferences to construct a directed multigraph. The preferences in the collection could be directly from preference judgments, derived from graded judgments, or they could include both. Fourthly, unlike traditional measures that compute relevance score by translating from preferences, Greedy PGC has a meaningful output score, which represents the maximum similarity between an actual ranking generated by users' preferences and an ideal ranking generated by a system.

There are many research directions that would be future work of this thesis. We have only examined the performance of Greedy PGC in terms of correlation with SERP preferences. In the future work, we hope to investigate the correlation between Greedy PGC and user click and satisfactions.

References

[1] Eugene Agichtein, Eric Brill, and Susan Dumais. Improving web search ranking by incorporating user behavior information. *SIGIR Forum*, 52(2):11–18, January 2019.

[2] R. Agrawal, A. Halverson, K. Kenthapadi, N. Mishra, and P. Tsaparas. Generating labels from clicks. In *2nd ACM International Conference on Web Search and Data Mining*, page 172–181, 2009.

[3] Nir Ailon, Moses Charikar, and Alantha Newman. Aggregating inconsistent information: Ranking and clustering. *J. ACM*, 55(5), November 2008.

[4] Azzah Al-Maskari, Mark Sanderson, and Paul Clough. The relationship between ir effectiveness measures and user satisfaction. SIGIR '07, page 773–774, New York, NY, USA, 2007. Association for Computing Machinery.

[5] Bonnie Berger and Peter W. Shor. Approximation alogorithms for the maximum acyclic subgraph problem. In *Proceedings of the First Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '90, page 236–243, USA, 1990. Society for Industrial and Applied Mathematics.

[6] F. J. Brandenburg and Kathrin Hanauer. Sorting heuristics for the feedback arc set problem. 2011.

[7] Ben Carterette, P. Bennett, and O. Chapelle. A test collection of preference judgments. 2017.

[8] Ben Carterette and Paul N. Bennett. Evaluation measures for preference judgments. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '08, page 685–686, New York, NY, USA, 2008. Association for Computing Machinery.

[9] Ben Carterette, Paul N Bennett, David Maxwell Chickering, and Susan T Dumais. Here or there. In *European Conference on Information Retrieval*, pages 16–27. Springer, 2008.

[10] Praveen Chandar and Ben Carterette. Using preference judgments for novel document retrieval. In *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '12, page 861–870, New York, NY, USA, 2012. Association for Computing Machinery.

[11] Praveen Chandar and Ben Carterette. Preference based evaluation measures for novelty and diversity. In *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '13, page 413–422, New York, NY, USA, 2013. Association for Computing Machinery.

[12] Olivier Chapelle, Donald Metlzer, Ya Zhang, and Pierre Grinspan. Expected reciprocal rank for graded relevance. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management*, CIKM '09, page 621–630, New York, NY, USA, 2009. Association for Computing Machinery.

[13] Mark D. Smucker Charles L.A. Clarke, Chengxi Luo. Evaluation measures based on preference graphs.

[14] Charles L. A. Clarke, Mark D. Smucker, and Alexandra Vtyurina. Offline evaluation by maximum similarity to an ideal ranking. In *Proceedings of the 29th ACM International Conference on Information amp; Knowledge Management*, CIKM '20, page 225–234, New York, NY, USA, 2020. Association for Computing Machinery.

[15] Charles L. A. Clarke, Alexandra Vtyurina, and Mark D. Smucker. Assessing top-k preferences, 2021.

[16] Charles L.A. Clarke, Alexandra Vtyurina, and Mark D. Smucker. Offline evaluation without gain. In *Proceedings of the 2020 ACM SIGIR on International Conference on Theory of Information Retrieval*, ICTIR '20, page 185–192, New York, NY, USA, 2020. Association for Computing Machinery.

[17] Kevyn Collins-Thompson, Craig Macdonald, Paul Bennett, Fernando Diaz, and Ellen M. Voorhees. Overview of the trec 2014 web track. In *In Proceedings of the 23rd Text REtrieval Conference (TREC '14)*, February 2015.

[18] Nick Craswell, Bhaskar Mitra, Emine Yilmaz, Daniel Campos, and Ellen M. Voorhees. Overview of the trec 2019 deep learning track, 2020.

[19] Maurizio Ferrari Dacrema, Paolo Cremonesi, and Dietmar Jannach. Are we really making much progress? a worrying analysis of recent neural recommendation approaches. In *Proceedings of the 13th ACM Conference on Recommender Systems*, RecSys '19, page 101–109, New York, NY, USA, 2019. Association for Computing Machinery.

[20] Jeffrey Dalton, Chenyan Xiong, and Jamie Callan. CAsT 2019: The Conversational Assistance Track overview. In *28th Text REtrieval Conference*, Gaithersburg, Maryland, 2019.

[21] Peter Eades, Xuemin Lin, and W.F. Smyth. A fast and effective heuristic for the feedback arc set problem. *Information Processing Letters*, 47(6):319–323, 1993.

[22] Guy Even, Joseph (Seffi) Naor, Baruch Schieber, and Madhu Sudan. Approximating minimum feedback sets and multi-cuts in directed graphs. In Egon Balas and Jens Clausen, editors, *Integer Programming and Combinatorial Optimization*, pages 14–28, Berlin, Heidelberg, 1995. Springer Berlin Heidelberg.

[23] Sean Falconer and Dmitri Maslov. Weighted hierarchical alignment of directed acyclic graph. 06 2006.

[24] H. P. Frei and P. Schäuble. Determining the effectiveness of retrieval algorithms. *Inf. Process. Manage.*, 27(2–3):153–164, April 1991.

[25] Google. Search quality evaluator guidelines. *Google*, October 2020.

[26] Martin Grötschel, Michael Jünger, and Gerhard Reinelt. Facets of the linear ordering polytope. *Mathematical programming*, 33(1):43–60, 1985.

[27] K. Hofmann, L. Li, and F. Radlinski. *Online Evaluation for Information Retrieval*. Foundations and Trends in Information Retrieval. Now Publishers, 2016.

[28] Kalervo Järvelin and Jaana Kekäläinen. Cumulated gain-based evaluation of ir techniques. *ACM Trans. Inf. Syst.*, 20(4):422–446, October 2002.

[29] Shubhra (Santu) K. Karmaker, Parikshit Sondhi, and ChengXiang Zhai. Empirical analysis of impact of query-specific customization of ndcg: A case-study with learning-to-rank methods. In *Proceedings of the 29th ACM International Conference on Information amp; Knowledge Management*, CIKM '20, page 3281–3284, New York, NY, USA, 2020. Association for Computing Machinery.

[30] Richard Karp. Reducibility among combinatorial problems. volume 40, pages 85–103, 01 1972.

[31] Diane Kelly and Jaime Teevan. Implicit feedback for inferring user preference: A bibliography. *SIGIR Forum*, 37(2):18–28, September 2003.

[32] Maurice George Kendall. Rank correlation methods. 1948.

[33] Youngho Kim, Ahmed Hassan, Ryen W. White, and Imed Zitouni. Modeling dwell time to predict click-level satisfaction. In *Proceedings of the 7th ACM International Conference on Web Search and Data Mining*, WSDM '14, page 193–202, New York, NY, USA, 2014. Association for Computing Machinery.

[34] P. Klein, Serge A. Plotkin, C. Stein, and Éva Tardos. Faster approximation algorithms for the unit capacity concurrent flow problem with applications to routing and finding sparse cuts. *SIAM J. Comput.*, 23:466–487, 1994.

[35] T. Leighton and S. Rao. An approximate max-flow min-cut theorem for uniform multicommodity flow problems with applications to approximation algorithms. In *Proceedings of the 29th Annual Symposium on Foundations of Computer Science*, SFCS '88, page 422–431, USA, 1988. IEEE Computer Society.

[36] C. E. Miller, A. W. Tucker, and R. A. Zemlin. Integer programming formulation of traveling salesman problems. *J. ACM*, 7(4):326–329, October 1960.

[37] Mark E Rorvig. The simple scalability of documents. *Journal of the American Society for Information Science*, 41(8):590–598, 1990.

[38] Tetsuya Sakai and Ruihua Song. Evaluating diversified search results using per-intent graded relevance. In *34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, page 1043–1052, Beijing, China, 2011.

[39] Tetsuya Sakai and Zhaohao Zeng. Which diversity evaluation measures are "good"? In *42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, page 595–604, Paris, France, 2019.

[40] Tetsuya Sakai and Zhaohao Zeng. Good evaluation measures based on document preferences. In *43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, page 359–368, 2020.

[41] Yunqiu Shao, Jiaxin Mao, Yiqun Liu, Min Zhang, and Shaoping Ma. Towards context-aware evaluation for image search. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR'19, page 1209–1212, New York, NY, USA, 2019. Association for Computing Machinery.

[42] Michael Simpson, Venkatesh Srinivasan, and Alex Thomo. Efficient computation of feedback arc set at web-scale. 10(3):133–144, November 2016.

[43] Nicola Stokes. Trec: Experiment and evaluation in information retrieval ellen m. voorhees and donna k. harman (editors) (national institute of standards and technology). *Computational Linguistics - COLI*, 32:563–567, 12 2006.

[44] Kosetsu Tsukuda and Masataka Goto. Query/task satisfaction and grid-based evaluation metrics under different image search intents. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '20, page 1877–1880, New York, NY, USA, 2020. Association for Computing Machinery.

[45] Vijay V Vazirani. *Approximation algorithms*. Springer Science & Business Media, 2013.

[46] Yining Wang, Liwei Wang, Yuanzhi Li, Di He, Tie-Yan Liu, and Wei Chen. A theoretical analysis of NDCG type ranking measures. *CoRR*, abs/1304.6480, 2013.

[47] William Webber, Alistair Moffat, and Justin Zobel. A similarity measure for indefinite rankings. 28(4), November 2010.

[48] Xiaohui Xie, Yiqun Liu, Xiaochuan Wang, Meng Wang, Zhijing Wu, Yingying Wu, Min Zhang, and Shaoping Ma. Investigating examination behavior of image search users. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '17, page 275–284, New York, NY, USA, 2017. Association for Computing Machinery.

[49] Xiaohui Xie, Jiaxin Mao, Yiqun Liu, Maarten de Rijke, Haitian Chen, Min Zhang, and Shaoping Ma. Preference-based evaluation metrics for web image search. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '20, page 369–378, New York, NY, USA, 2020. Association for Computing Machinery.

[50] Xiaohui Xie, Jiaxin Mao, Yiqun Liu, Maarten de Rijke, Yunqiu Shao, Zixin Ye, Min Zhang, and Shaoping Ma. Grid-based evaluation metrics for web image search. In

The World Wide Web Conference, WWW '19, page 2103–2114, New York, NY, USA, 2019. Association for Computing Machinery.

[51] Xiaopeng Yang, Yongdong Zhang, Ting Yao, Chong-Wah Ngo, and Tao Mei. Click-boosting multi-modality graph-based reranking for image search. *Multimedia Syst.*, 21(2):217–227, March 2015.

[52] YY Yao. Measuring retrieval effectiveness based on user preference of documents. *Journal of the American Society for Information science*, 46(2):133–145, 1995.

[53] Emine Yilmaz, Javed Aslam, and Stephen Robertson. A new rank correlation coefficient for information retrieval. pages 587–594, 01 2008.

[54] Fan Zhang, Ke Zhou, Yunqiu Shao, Cheng Luo, Min Zhang, and Shaoping Ma. How well do offline and online evaluation metrics measure user satisfaction in web image search? In *The 41st International ACM SIGIR Conference on Research amp; Development in Information Retrieval*, SIGIR '18, page 615–624, New York, NY, USA, 2018. Association for Computing Machinery.

APPENDICES