Trinity College Dublin
Coláiste na Tríonóide, Baile Átha Cliath
The University of Dublin

School of Computer Science and Statistics

Personalized Type-based Facet Ranking For Faceted Search

Esraa Ali Abdelmonem

15337092

January 2022

A PhD dissertation submitted in partial fulfilment
of the requirements for the degree of
PhD (Computer Science)

Supervised by

Séamus Lawless, Annalina Caputo, and Owen Conlan

Declaration

I hereby declare that this project is entirely my own work and that it has not been submitted as an exercise for a degree at this or any other university.

I have read and I understand the plagiarism provisions in the General Regulations of the University Calendar for the current year, found at `http://www.tcd.ie/calendar`.

I have also completed the Online Tutorial on avoiding plagiarism 'Ready Steady Write', located at
`http://tcd-ie.libguides.com/plagiarism/ready-steady-write`.

Signed: _____ Date: _____

Abstract

Faceted Search Systems (FSSs) have gained prominence as one of the dominant search approaches in vertical search systems. They provide facets to educate users about the information space and allow them to refine their search query and navigate back and forth between resources on a single results page.

Type-based facets (aka t-facets) help explore the categories associated with the searched objects. In a structured information space, t-facets are usually derived from large hierarchical taxonomies. When the information available in the collection being searched increases, so does the number of associated t-facets. This makes it impractical to display at once, the entire t-facet taxonomy to the user.

To tackle this problem, facet ranking is implemented in the FSS. Ranking methods can take advantage of the information structure, the textual queries issued by the users, or the user logs. However, existing methods neglect both the hierarchical structure of the taxonomies, and hence of the t-facets, as well the user's historical preferences. As a consequence, users obtain a t-facet list that is difficult to read and irrelevant to their interests.

This thesis focuses on the task of personalizing t-facet ranking in precision-oriented FSSs. It investigates to what extent personalizing t-facet ranking, using user historical feedback, can minimize the user effort to reach the intended search target. This work proposes a two-step approach to solve this problem. The first step scores each individual leaf-node t-facet. In the second step, this score is used to re-order and select the sub-tree to present to the user. The final ranked tree reflects the t-facet

relevance both to the query and the user profile.

For the first step, three scoring methods are developed: A probabilistic, a Vector Space Model (VSM), and a Deep Neural Network (DNN) scoring method. These methods combine different types of information in different ways to recommend the most relevant t-facets to the user. A common element to all of them is the individual user profile collected from the previous user's ratings in the system. The second step suggests three strategies to utilize this generated score in order to produce the final t-facet sub-tree to the user. The strategies group the relevant t-facets by their parent-nodes and order them by aggregating the t-facet scores from the earlier step.

Whilst evaluation protocols have been developed for the general facet ranking, the problem of personalizing the facet rank, based on user profile, has lagged behind due to the lack of appropriate datasets. To fill this gap, this thesis introduces a framework to reuse and customise existing real-life data collections. The framework outlines the eligibility criteria and the data structure requirements needed for this task. It also details the process to transform the data into a ground-truth dataset. We apply this framework to two existing data collections in the domain of Point-of-Interest (POI) suggestion: TREC-CS 2016 and Yelp Open Dataset.

In order to assess the performance of the proposed approach, the proposed framework combines the widely adopted user-simulation model and metrics proposed in the INEX 2011 Data Centric Faceted Search task [1]. The evaluation approach aims to capture the user's effort required to fulfill their search needs, by using the ranked t-facet tree. Our experiments have found that the proposed DNN based scoring methods significantly minimize the number of actions the user need to perform in order to reach the intended search target. The evaluation results also demonstrated that using different tree construction strategies have a significant impact on the same number of actions metric. We conclude that the proposed personalized approach leads to better t-facet rankings and minimizes user effort.

Keywords: Type-based Facets, Faceted Search, Facet Ranking, and Personalization.

To my support and backbone: My husband Ahmad, my parents Ali and Amnah, my kids Omar and Sara. For you I live.

To my inspirations: My grandmother Kawkab, my aunt Fayza, my great aunt Awatef, my sis Mona, and to all the wonderful women out there pushing the limits set for them. Keep shining ladies!.

To my mentors: late Prof. Shay Lawless, late Prof. Mustafa Ghanem, Prof. Waleed Youssef, Dr Amr Ghonim. I promise to pass the torch.

Acknowledgements

First and foremost, I would like to praise and thank Allah, the almighty, who has granted me countless blessings, knowledge, and opportunities, so that I have been finally able to accomplish the thesis in these difficult times.

I was fortunate to be given the opportunity to pursue my research passion by Prof. Séamus Lawless. He believed in my capabilities and gave me the chance to work amongst excellent researchers. His patience and deep discussions opened my mind. He mentored me during the hard phase of articulating a research point and a PhD plan. He taught me to aim high, and to follow my passion even in research related decisions. He showed me that a person can be a great academic, a successful manager and a very kind human being at the same time. The tragic accident in Everest was a shock to everyone who knew him, he left a place which will never be filled. I wished I could tell him how much he changed my life, and that I am forever in his debt. Also many thanks to Prof. Owen Conlan for stepping in and agreeing to be my supervisor in spite of his busy schedule and other commitments. Owen's continuous support and encouragement built my confidence. His insights and feedback about my work helped me focus, organize my thoughts and shape this thesis.

I would like to say a special thank you to my co-supervisor Annalina Captuo for her unlimited and unconditional support and understanding. Her guidance and friendship was vital to me to get through this tough journey. I was lucky that Annalina joined the team in the first year of my PhD., since then she became the person I go to for discussing any thoughts related to my work and even any crazy idea, she would listen, think with me and eventually improve it to something meaningful. No words can describe how grateful I am for her, she is always available when I need her. Her passion about research inspires me, she is the kind of academic I look up to. Thank you Annalina.

I also would like to extend my thanks to my beloved husband Ahmad Abdelghany for enduring so much with me during the PhD. He is my biggest supporter, always there motivating and pushing me to do better all the time. During many PhD ups

and downs, he believed in me even when I was about to give up. It is because of him I was able to reach this milestone in my life. I am forever grateful. I love you so much.

I would like to express my sincere gratitude to my parents Ali Hassan and Amnah Kamel. They didn't spare any effort to help me during my studies. Their unconditional love and care mean everything to me. My heart can't thank them enough for giving me such a strong foundation and helping me be the person I am today. Also thanks to my sister Mona, my brothers Mohamed and Ahmed, all my cousins and the big family who always encouraged and supported me throughout my career. Big thank you to my amazing childminder Yasmina Saf. Because of her I could focus on my research knowing that my kids are well taken care of. She beared with me through difficult COVID-19 times. I truly appreciate all your effort Yasmina.

Thank you my PTUE-IR research team for all the support and advice : Dr Brendan Spillane, Dr Gary Munnelly, Dr Mostafa Bayoumi, Dr Anirban Chakraborty, Dr Yu Xu and Hao Wu. Much of my experimental work would have not been completed without the assistance of ADAPT's staff, especially the technical support team (Thank you Graziano D'Innocenzo and Cormac McKenna). Also thanks to my colleagues and former ADAPT members: Dr Rami Ghorab, Dr Retno Aulia, Dr Ramisa Hamed.

Special thank you to my dearest friend Dima Mahmoud for being there when needed, you are a wonderful person, an excellent researcher and you deserve everything good. Also thanks to my friend Fatma Taher for the technical discussions. Thanks to my Egyptian friends in Ireland (Basma, Engy, Hanan, Salma, Shereen, and many others) for making my stay in Ireland much easier, you are my second family. Thank you my friends from Egypt (Ameera, Amira, Khadiga, Eman, Shaimaa, Noha, Nehal, and Nahla) for the continuous support and encouragement.

Thank You All.

Acronyms

DL	Deep Learning
DNN	Deep Neural Network
EL	Entity Linking
ESS	Exploratory Search System
FSS	Faceted Search System
HCI	Human Computer Interaction
IR	Information Retrieval
KB	Knowledge Base
KNN	K-Nearest Neighbour
LM	Language Model
LTR	Learning To Rank
ML	Machine Learning
NLP	Natural Language Processing
PIR	Personalized Information Retrieval
POI	Point of Interest
UI	User Interface or Graphical User Interface
VSM	Vector Space Model

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Faceted search is one of the mainstream search paradigms in vertical search engines. In addition to famous look-up search systems, faceted search systems (also known as faceted browsing or faceted navigation) provide an alternative way for the user to navigate through the search space. It is the de facto search approach for many domain specific search engines like e-commerce, e-tourism and other websites. This thesis focuses on how personalization can play a role in improving the faceted search process.

This chapter starts with a brief background about faceted search systems and what differentiate them from other search systems in section 1.1. It also defines facets, their different categorization and how they are used. Next, section 1.2 draws the motivation behind the focus on type-based facet ranking, summarizes how other researchers approached this problem, and identifies the research gap addressed by this study.

The main research question and objectives pursued by this thesis are described in section 1.3. Then, the methodology followed to accomplish those objectives and answer the research question, is illustrated in section 1.5. Section 1.6 details the contribution and deliverables of this PhD work. Finally, the rest of the thesis structure

is demonstrated in section 1.7.

1.1 Background

In Information Retrieval (IR) research Faceted Search Systems (FSSs) refer to a family of systems which summarize, organize, and aggregate search results in one page making it easier to grasp the information space without going back and forth between web pages. The primary distinction between faceted search and other forms of web search is that users can explore the information space through facets. Facets are attributes or meta-data that describe the underlying content collection.

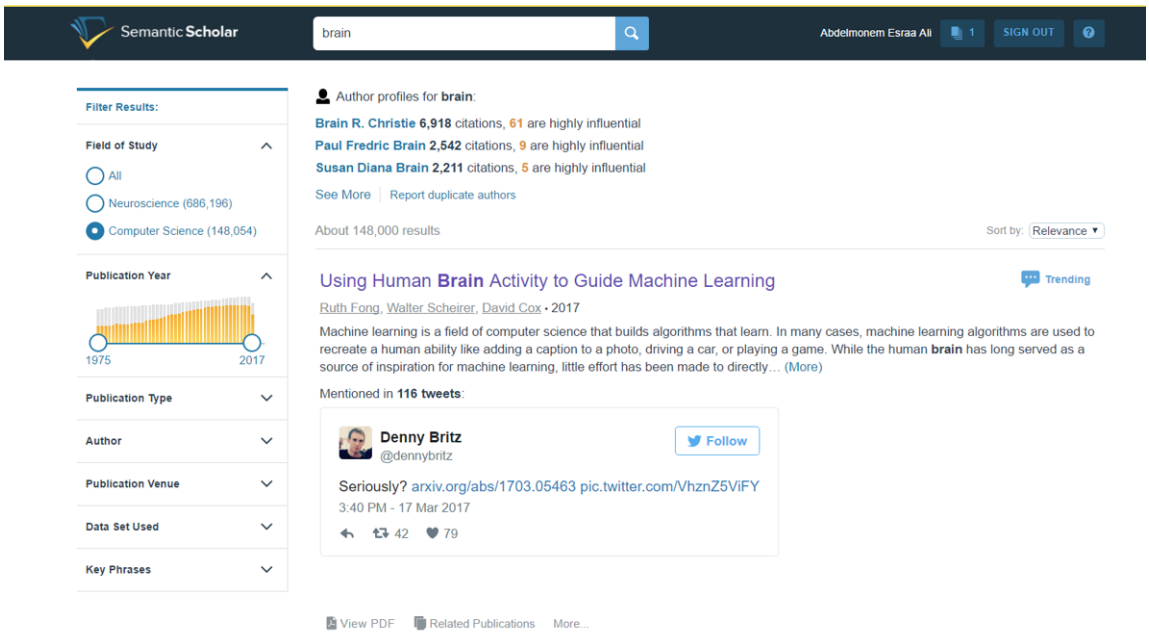


Figure 1.1: Faceted search interface example: Semantic Scholar

Figure 1.1 shows an example of a faceted search interface called Semantic Scholar [2]. Using facets on the left side and top panes, the interface aggregates and summarizes information to the user on a single page. By doing this, the system educates the user about the information domain rather than giving back one specific answer. From the page, users can learn about the top authors in the field and the amount

of research published over the years which mentions the search term(s). It also gives some context by mentioning what is trending now in this research area.

In faceted search, the searcher can use the filters (facets) to navigate the data and learn more about the research area in general. For example, if the user selected a filter value for a specific year, the interactive interface will narrow down the search results to show only research published in this specific year.

Similarly, if a specific conference was selected, the results would be narrowed to only include papers published at that conference. If the user deselects the conference filter, the system reverts to a broader set of search results. All these narrowing and widening processes usually happen on the client side, without the need to change the page or re-run the search.

In an information space that is structured, facets are either extracted from relationships between objects, in this case they are called *Property-based facets* (***p-facets***), or they are extracted from the types of objects, in which case they are called *Type-based facets* (***t-facets***) (e.g. the values of `type` or `isA` relationships) [3]. Systems vary in their use of facets, some use a single type of facet [4, 5, 6, 7], others mix the two types [8]. Usually this is done by presenting the type-based facets first, followed by the property-based facets [9, 10, 11]. In the Semantic Scholar FSS, examples of t-facets are the 'Field of Study' and the 'Publication Type' facets. On the other hand, the 'Publication Year', 'Author', and 'Publication Venue' are examples of the p-facets.

Type-based facets typically have a hierarchical taxonomic structure, which enables users to filter the search results by choosing from a predetermined set of categories. Hierarchical taxonomies of types are derived from ontologies by exploiting the `subClassOf` relationships. This PhD investigates methods to personalize the ranking of type-based facets in order to deliver a personalized search experience. The motivation for this research scope is outlined in the next section.

1.2 Motivation

As the magnitude of data in a collection increases, the number of facets and their values become impractical to display in a single page. Providing users with too many facets has been shown to overwhelm and distract them [12, 13, 14].

Existing faceted browsers overcome this problem by either displaying a small number of facets, and making the rest accessible through a "more" button, or by displaying only the facet titles without the values, and if the user is interested in a facet, they can click on the title to view its values. In either case, ranking the top facets is required as it guides the searcher in understanding the main aspects of the information space being explored.

Facets can be generated from both structured, semi-structured, or unstructured data sources. FSS that operate on unstructured data perform a facet generation step before applying facet ranking. The quality of facet generation algorithms can impact the facet ranking step. In the case of structured data, facets are directly collected from objects attributes (more details about facet generation in chapter 2, section 2.2.3). In order to evaluate the facet ranking step in isolation from the facet generation step, this PhD research focuses on the ranking of facets collected from the structured parts of the data.

Existing facet ranking methods rely on scores derived from the information structure, such as attribute frequencies, navigation cost models, textual queries or click logs [3]. However, the importance of a facet is not necessarily conveyed by its frequency of occurrence in the dataset. For example, in an information space describing people, presidency-related facets can occur very infrequently, yet if someone has served as president, this is more significant than other person-related facets.

Ranking facets goes beyond the choice of which facets can partition the data more effectively. Instead, it is more concerned with educating the user about the information space by identifying the key facets that describe it in the best way possible. Finally, ranking based solely on usage logs does not effectively reflect individual user

interests, or the context of their current task, and usually has a slow response to changes in users' behaviour [3, 15, 16].

While the majority of existing personalized facet ranking methods consider the user preferences, they do not distinguish the special t-facet case from other p-facets during the ordering process [14, 17, 18]. In faceted search systems which exploit multiple types of documents [1], it is important to prioritize and focus on the relevant type-based facets first. This is especially true when the types of documents come from a large multilevel hierarchy. This will encourage the user to filter the results by their type first and make it easier to group and rank the rest of the property-based facets (if they exist).

As far as is known, there is an existing work which personalizes t-facets, which uses only current session interaction [4]. Neither the historical user interests nor the hierarchical nature of t-facets are considered by this approach.

This thesis aims to portray the different ways in which personalization can be applied in the type-based facet ranking problem. The deemed personalization is based on individual user history which is not covered by earlier literature. It is also crucial for the intended ranking method to include query relevance as well as other importance signals in the ranking process. It should also take into account the taxonomy from which the t-facets are derived.

The final output t-facet tree in precision-oriented FSS seeks to help the users in finding their search target with minimum effort. This effort is usually measured by the number of clicks the users had to perform in order to reach their search target. It can also be measured by the number of t-facets and documents the user had to scan before reaching the intended search target [1].

[1] In the scope of this thesis, the term 'documents' is used to refer to the information objects being searched. According to the FSS domain, documents can be places, web pages, products, books or images ... etc.

1.3 Research Question and Objectives

The research question pursued in the PhD work is:

> ***RQ:*** Can personalizing type-based facet ranking according to user historical feedback minimize the effort needed by users to fulfill their search needs?

The users' effort is measured by the number of actions they had to perform on a ranked t-facet list in order to reach the intended search target. The main hypothesis, to be proven during evaluation, is that customizing the ranking method according to user interests will significantly improve and facilitate the process of search target finding. In order to answer this research question and prove the stated hypothesis, the following research objectives were targeted:

1. To review the state of the art research related to faceted search systems in general, with a focus on existing facet ranking approaches.
2. To investigate state-of-the-art FSS evaluation frameworks, and the most commonly used techniques and metrics to evaluate facet ranking approaches.
3. To define criteria for dataset appropriateness and its needed structure.
4. To choose a domain task and collect suitable data for this task based on the defined criteria.
5. To identify, implement and examine existing baseline facet ranking methods, and analyse how they perform on the chosen dataset(s) and evaluation framework.
6. To exploit several personalization strategies from Personalized Information Retrieval (PIR) literature that can be applied to t-facet ranking.
7. To evaluate the proposed ranking approach, using well established evaluation methods and metrics.
8. To compare the proposed approach against state of the art baselines, and reflect

on the most effective system.

An assumption of this research is that the ranking of facets occurs during the initial population of the result page, and that they are not reshuffled during the navigation process unless the user submits a new query, which will re-initiate the facet ranking step.

1.4 Key Research Challenges

1.4.1 Establishing T-Facet Relevance

The purpose of this work is to investigate solutions for the problem of ranking type-based facets generated from structured data. Normally, this data is provided by knowledge graphs. They have the advantage of containing high quality structured ontologies for the t-facets. However, with the increasing size and complexity of these collections, the task of deciding which t-facets should be manifested to the user, and in which order, becomes more difficult.

In addition to that, relevance of the t-facet is subjective. It depends on several factors, the users' interests, its relevance to the input query and current search contexts, as well as its general importance in the collection. All these factors contribute to the t-facet relevance and should be considered by the ranking approach.

Moreover, from the personalization perspective, relevance of the t-facets varies not only from one person to another, but also for the same person, it changes from one situation to another. Users' knowledge, interests, and therefore search needs, evolve with time. Keeping track of an updated user profile and reflecting that in the search results adds difficulty to the problem.

1.4.2 Maintaining the Multilevel Taxonomy Structure

This is especially true in the t-facets case. When the t-facets originate from a large multilevel taxonomy, the difficulty of the ranking process increases; it goes beyond

computing a score for each t-facet type, and it involves other decisions, like which levels of the taxonomy need to be displayed to the user. Also, the ranking needs to consider how to order and rank the type-based facet preserving the original taxonomy and without confusing the user.

1.4.3 Avoid Adding Complexity to the FSS

The facet ranking phase is usually triggered after the search engine retrieves a set of relevant documents as a response to an initial query submitted by the user. This is a document-level score generation phase that often requires heavy computation depending on the underlying IR method used. Adding more complex processing and computation for the facet ranking layer is not desirable. Instead, a light and effective method is crucial as it impacts users' experience and their perception of the system. At the same time, the ranking method should effectively aid the searcher in narrowing and focusing the information space to retrieve the most relevant results according to the users' interests and desires.

1.4.4 Deciding on a Search Task and its Objectives

Moreover, defining the search task is pivotal. It affects how the ranking happens, i.e. whether it should favor covering as much information as possible by giving the user a broad idea about the topic, or favor minimizing the navigation time to enable the searcher to find the target resource as quickly as possible. In addition to these two, there are many other scenarios where the ranking objective will vary according to the domain and the search task. Focusing on a well-defined search task and objective is fundamental for the development of a proper t-facet ranking method.

1.4.5 Evaluating Personalized T-Facet Ranking

The most challenging part of this research was to find an appropriate well-established evaluation methodology. This gives confidence in the interpretation of results. Previous studies of personalized facet ranking have not dealt with the special case of

type-based facets. This creates a set of challenges to be faced while evaluating the proposed approach: 1) The lack of t-facet ranking baseline methods. 2) As a result, the area also lacks existing bench-marked datasets that fit the purpose of this research. 3) Existing evaluation metrics are also developed for the generic facet ranking; they do not consider the hierarchical nature of the t-facets.

1.5 Research Methodology

This research was initiated by performing a comprehensive review of literature related to FSS, how search occurs, what are the key components in a typical FSS, how facets are generated and ranked, and how users interact with them. The lack of personalized FSS based on user interests became evident.

After a couple of experiments on entity attribute ranking and `isA` relationship triples scoring [2], an important lesson was learned; different ranking approaches are needed for each type of facet. Whilst the information structure might be sufficient for property-based facets, they cannot convey neither the type importance nor the user relevance on its own without any additional data. Another specialized ranking approach for t-facets was needed. The focus on personalized type-based facet ranking was then decided.

In the investigated existing literature, only few published works could be found about personalized facet ranking methods based on user's previous ratings. Moreover, the hierarchical nature of the t-facets was not addressed by prior research. In particular, the specific case of personalized type-based facet ranking remains unexplored by previous research.

To solve this problem, this study proposes a novel personalized type-based facet ranking approach. The proposed approach functions over two consecutive stages. The first stage generates a personalized relevance score for each t-facet at the end level of

[2]Published in [19] and [20] but not included in PhD deliverables as they are not directly related to the thesis topic. However, they were useful to the formation and development of the ideas presented in this thesis.

the taxonomy. This step addresses the challenge of establishing t-facet relevance presented in section 1.4.1. Then, the second stage decides how this generated score can re-arrange and re-build the relevant t-facet tree to be rendered to the user. For that purpose, several tree construction strategies are formulated. The strategies respect and preserve the original type taxonomy structure while determining how ancestor-level types are going to be ordered and portrayed to the user. The tree construction step tackles the second research challenge regarding maintaining the structured taxonomy introduced in section 1.4.2.

At the score generation stage, three personalization alternative methods are suggested. The three methods are derived from known IR ranking techniques. The first method is based on a probabilistic pseudo-relevance approach. The second is based on a vector space model approach. Finally, the third is a learning to rank approach using Deep Neural Network (DNN). All methods attempt to model a user profile from their previous ratings. In addition to that, the aforementioned methods also incorporate the relevance score of the document, to which the t-facet belongs, into the t-facet's final generated scores. This document score is generated after the underlying documents are ranked by the search engine, and before the facet ranking phase takes place. The developed methods avoid complex computations which overcomes the third challenge described in section 1.4.3.

Most existing FSSs employ the same ranking methods for different types of facets (including t-facts). Since there is a lack of methods focusing on the t-facet ranking, baselines were obtained from existing facet ranking methods. Key personalized and non-personalized methods were implemented, analysed and compared against the proposed approaches.

Although the literature presents a wealth of research in FSS, this area still lacks standard datasets with ground truth for facet ranking. This problem is even more relevant for personalized facet ranking tasks. To overcome this problem, researchers either proceed with a task-based user evaluation, or use user simulation approaches and customize existing IR datasets to solve the facet ranking problem.

Experiments in this PhD concentrate on ranking type-based facets in the domain of tourism, specifically venue suggestion (also known as Point of Interest suggestion). This domain is appropriate for evaluation of the proposed approach for several reasons: 1) The problem is personalized in nature, as the ranking process aims to guide the user to find interesting places to visit; 2) venue types usually belong to a rich multilevel taxonomy and their selection reflects user interests; 3) availability of benchmark IR dataset like TREC-CS, which we use as an preliminary evaluation dataset [21]; and 4) the availability of large scale real data, like Yelp Open Dataset, where the scalability and generalizabilty of the proposed approaches can be tested. Deciding on a well articulated search task and its objects meet the fourth research challenge addressed by this research described in section 1.4.4.

It was essential to carefully review the existing FSS evaluation frameworks, and the challenges raised by evaluating personalized IR systems in general. Evaluating personalization in a user-based scenario requires expensive large scale user study. To avoid that, this thesis follows a simulated user evaluation. It also introduced a dataset customization framework to transform existing personalized IR collections like TREC-CS and Yelp to fit the purpose of t-facet ranking evaluation.

This research follows a widely adopted user interaction simulation model for evaluation. This includes a set of metrics developed for facet ranking by a known IR evaluation task (INEX Data Centric Faceted Search track [1]). The evaluation strategy builds upon the same user interaction model, it derives facet relevance judgment from documents ranking relevance judgment, which makes it a perfect fit for our evaluation using TREC-CS and Yelp datasets.

Having a well founded evaluation framework and well established benchmark datasets is crucial to drawing conclusive findings from experimental outcomes. The chosen evaluation framework also helps in facing the final research challenge listed in section 1.4.5. From a software engineering perspective, the implementation of the baselines, the evaluation metrics, and proposed methods went through several cycles of revision to make sure any errors or bugs were fixed. Experiments were executed

for all combinations of scoring methods and tree construction strategies. Then the best performing methods were compared against chosen baselines.

After conducting the experiments, reported results have shown that DNN-based scoring methods with their variations, when compared to the best non-personalized method, significantly minimizes user effort needed to fulfill their research needs. These results outperform existing personalized baselines. Using this systematic thorough research methodology, we concluded that different t-facet tree construction strategies influence the evaluation metrics, and therefore they should be carefully chosen according to the search task and its objectives. It was also found that longer user profiles improve the personalization methods performance, and that personalization methods crafted specially for t-facet ranking can effectively minimize user effort needed to reach the intended search target.

1.6 Research Contribution

The main contribution of this work is the introduction of a personalized ranking approach for type-based facets. In addition to that, this study presents a dataset customization framework that can be used to create bench-marked datasets for this task evaluation. The contribution area is new and not explored by earlier research. Detailed contribution is outlined below:

1. This study proposes the first personalized type-based facet ranking approach. The approach has two main steps, the first step assigns a relevance score to each t-facet. The second step is concerned with using the generated scores to re-construct a tree for the relevant t-facets to be presented to the user.

2. For the first step, three personalized scoring methods are provided. The scoring methods incorporate the user's historical ratings as well as query relevance signals into the final score. The methods are:

   (a) Probabilistic scoring method, which utilizes user historical ratings and query relevance scores to compute the final t-facet score.

(b) Learning to rank scoring method, it combines user profile, query relevance, and collaborative filtering signals. These are used to train a Deep Neural Network LTR model which predicts the t-facet score.

(c) Vector space model based scoring methods, which utilizes the t-facet embeddings and an improved version of Rocchio formula to build a user profile vector and use it to compute the relevance of the t-facet.

3. This is the first study to highlight the need for a separate tree re-construction step to present the final ordered t-facet tree to the user. We suggest several construction strategies which utilize both the original taxonomy structure as well as the t-facets' score generated from earlier step.

4. The development of a framework for personalised t-facet ranking, which defines the criteria and structure of the needed dataset. Moreover, the framework specifies how to convert existing IR benchmarks and existing real life datasets to fit the purpose of t-facet ranking task evaluation.

5. Using the developed framework, two personalized t-facet ranking benchmark datasets are introduced in this work and can be utilized by researchers working in this area.

6. This research also adapts existing FSS simulated users evaluation approach to the t-facets case and formalizes how such a t-facet ranking approach can be evaluated.

In order to gain a deeper understanding of the proposed methods and their characteristics, the evaluation compares the proposed scoring and tree construction methods against the state-of-the-art facet ranking baselines.

1.6.1 Publications

Below is the list of current publications based on the experimental work done in this thesis:

1. The probabilistic scoring method and its results on the TREC-CS dataset are reported in this publication:
   Esraa Ali, Annalina Caputo, Séamus Lawless and Owen Conlan. A Probabilistic Approach to Personalize Type-based Facet Ranking for POI Suggestion. In Proceedings of 21st The International Conference on Web Engineering (ICWE 2021).

2. This publication covers the vector space model and its results on the TREC-CS dataset:
   Esraa Ali, Annalina Caputo, Séamus Lawless and Owen Conlan. Personalizing Type-based Facet Ranking using BERT Embeddings. In Proceedings of SEMANTiCS (SEMANTiCS 2021).

3. The details of the dataset customization framework and the results of the two previous methods on both the TREC-CS and Yelp datasets are included in this submission:
   Esraa Ali, Annalina Caputo, Séamus Lawless and Owen Conlan. Dataset Creation Framework for Personalized Type-based Facet Ranking Tasks Evaluation. In Proceedings of the Twelfth International Conference of the CLEF Association (CLEF 2021).

4. This submission covers the third and final scoring method based on DNN LTR:
   Esraa Ali, Annalina Caputo, Séamus Lawless and Owen Conlan. Where Should I Go? A Deep Learning Approach To Personalize Type-based Facet Ranking for POI Suggestion. In Proceedings of International Conference of Web Information Systems Engineering (WISE 2021).

5. Doctoral consortium paper:
   Esraa Ali. Dynamic personalized ranking of facets for exploratory search. In Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '17, pages 1379–1379, New York,NY, USA, 2017. ACM.

Additional in progress publications include:

- A Journal submission is being prepared. It will summarize the overall thesis outcomes with a focus on the t-facet tree construction strategies and how they impact the evaluation metrics.

- Resource paper to document and give access to the two generated datasets suitable for the personalized t-facet ranking evaluation, which will be made available as part of this research resources.

1.7 Thesis Organization

The overall structure of this thesis takes the form of six chapters. This chapter introduced the research problem, its scope, the motivation behind it and the several challenges faced while tackling this problem. It also outlines the main research question that this work aims to answer. The objectives were presented along with the contribution and deliverables by this thesis.

The rest of the thesis is organized as follows, the state-of-the-art research is surveyed in chapter 2. The chapter commences by introducing faceted search systems in section 2.1. It gives a brief overview about faceted search systems; how the search process happens, the data structure, and the information needs within FSSs.

Because of its impact on the facet ranking process, section 2.2 is dedicated to giving a closer look at the facets, what they are, how they are classified as well as the facet generation step and the common methods used for it. Section 2.3 reviews methods for facet ranking including personalized and non-personalized ones. The chapter also examines existing FSSs evaluation strategies and metrics in section 2.4. It concludes with current and open research directions in facet ranking and highlights the research gap addressed by this thesis.

The third chapter presents the new proposed approach in detail. It builds upon lessons learned from the literature review and outlines the design of the proposed

methodology to fulfill the objectives targeted by this research. The overall idea of the methodology is explained in the first section 3.2, this also introduces the requirements of the underlying data and the formal notations used in the remaining of the chapter.

The approach consists of two main steps, first a t-facet scoring detailed in section 3.3. Secondly, a t-facet tree construction step demonstrated in section 3.4. For each step a number of alternative methods are suggested. The chapter concludes with a discussion about the impact of each step, and how they interact together, in section 3.5.

Next, chapter 4 provides the implementation details and the parameters used in the experimental design. Section 4.1 is dedicated to the dataset customization framework. The section states the criteria followed to chose the datasets, their structure, statistics, and how they were customized to fit the t-facet ranking task. The baseline methods used to compare our proposed method are provided in section 4.3. In order to assess the proposed method, the thesis adopts a famous well established evaluation method, and a number of metrics used to evaluate facet ranking. An in-depth look into the adopted evaluation method, metrics, and how they are interpreted can be found in section 4.4.

The carried out evaluation and its outcomes are reported in chapter 5. The first section 5.1 gives an overview about the difficulties faced during the implementation of the proposed approach, the baselines, and the evaluation method. Section 5.2 reports the results for the proposed scoring and tree construction methods. The following section 5.3 compares the best performing methods with existing personalized and non-personalized baselines. In the same section the significance of the reported results are investigated. The limitations of the proposed approach and the adopted evaluation method are discussed in section 5.4. The last section in the chapter 5.6 summarizes the key findings of the experimental results.

Finally, the conclusions drawn from the conducted experiments are summarized in chapter 6. Section 6.1 demonstrates how the experimental results answered the research question posed in this thesis and details how the research objectives are met

by this work. The following section 6.2 elaborates on the possible application areas of this PhD and its industrial impact. Future work and open research directions are suggested in section 6.3. This thesis also includes appendix A1 which reports the detailed results of the experiments.

Chapter 2

Literature Survey

Faceted search is an area that is heavily studied in literature. This chapter gives an overview about faceted search systems in general, what they are, how they operate and examples of existing systems. In addition to that, key FSS aspects relevant to this research are covered in section 2.1, like information need, user interaction, the search process, and underlying data structure. Other FSS aspects, like query understanding, data indexing, and visualization are important for the user experience, but they are outside the scope of this research as they have less impact on the facet ranking process.

The following section 2.2 focuses on introducing facets, their definition, and how they are classified in literature. It also draws attention to the facet generation phase in more detail due to its impact on the facet ranking process. Section 2.3 is dedicated to reviewing existing facet ranking approaches in literature. They can be broadly classified into personalized and non-personalized ranking methods. The section provides discussion of the benefits and drawbacks of each approach.

The next section in this chapter (Sec. 2.4) covers the most commonly adopted evaluation strategies in FSS, and focuses on how facet ranking can be evaluated aside from other FSS components. It also discusses how the domain and search task choice affect the evaluation process.

Finally, this survey is concluded with summary of the FSS's literature in section 2.5. It also highlights the research gap addressed by this thesis in section 2.6.1.

2.1 Faceted Search

Faceted search refers to a family of look-up systems which enables users to explore, digest, analyze and navigate through multi-dimensional complex information spaces [22]. It is a popular easy to use interaction paradigm for users in which they use common metadata or attributes (facets) to browse the information objects being searched. Several studies found that users like faceted search systems, they found them intuitive and easy to use [13, 23].

The browsing paradigm emerged in literature as early as 1930s. It was originally based on facet analysis theory introduced by mathematicians in information sciences. This theory was further developed and widely adopted later in information retrieval field, where in the last two decades it gained popularity [24]. Currently, faceted search is the dominant approach used in vertical search domains like e-commerce websites, tourism websites and digital libraries. Terminology wise, faceted search is also mentioned in literature as faceted browsing, faceted navigation, multifaceted search, or guided navigation [22, 24].

Formally, Tzitzikas et al. [3] defines faceted search as:

> *"A session-based interactive method for query formulation (commonly over a multidimensional information space) through simple clicks that offers an overview of the result set (groups and count information), never leading to empty results sets."*

Figure 2.1 shows an example of a faceted search system for Wikipedia that utilises DBpedia relationships [25]. The system is called Faceted Wikipedia Search. The interface displays the facets and their values on the left side of the screen. Facets based on the type of the searched objects are displayed first in a separate box (Item Type Selection), followed by other groups of facets.

Figure 2.1: Faceted Wikipedia Search

In Faceted Wikipedia Search, facet-values are ordered by count. The interface presents the three facet values with highest count for display, and makes the remaining values for each facet available through a 'more' button. The top panel in the middle helps to keep track of the selected facets, whilst the middle of the page shows a ranked list of resources that satisfies the currently applied facet filters or conditions.

2.1.1 The Search Process

The search process in a typical FSS involves a number of iterative steps [22], in which the searchers can:

1. Type or refine a search query, or
2. Navigate through multiple, independent facet hierarchies that describe the data by drill-down (refinement) or roll-up (generalization) operations.

In FSS which support search queries (for example [26] and others), a typical usage scenario starts with the user entering a search query. The system processes this query to find a list of relevant resources. These resources can be RDF entities, documents, or multimedia objects, depending on the underlying information representation. In case the FSS does not support keyword search queries, initially all information space is considered relevant and the first page is populated with the same starting documents each time.

Regardless of the nature of the data, FSSs assume that the data have attributes in common. The next step is to generate a set of common facets and their values. The facet-values are then collected, grouped, counted, and can be organized into hierarchies. For example, if the facet-values are dates, the system can try to find the best grouping by year, month, or day.

This organization gives a better understanding of the data distribution, and it is useful for minimizing the drill down time. After that, an appropriate label for each facet and its values are produced and used in the display. At this stage, the initial page is populated, and from this point forward the user starts navigating and

exploring the data [3, 6, 10, 11, 12, 15, 23, 27, 28, 29, 30].

2.1.2 User Interaction Model

The user starts interacting with the FSS after the first page is populated with data. The provided user interface allows the user to select and deselect facets, and filters the search results according to the current set of selected facets. This interface allows the user to add more than one facet in order to narrow down or restrict the search results. Users can also remove one or more facets from the current selection set in order to broaden or expand the search results. Providing the users with means to select and deselect multiple facets enables them to build complex filtering conditions to satisfy their complex search needs [3, 22].

To avoid the user from feeling lost in the system, the current set of applied filters are displayed at all times. This is important in allowing the user to identify the current search state. At the same time, it also supports the users in deciding which is the next action to be performed. As soon as the searchers perform any action on the facets or search results, the FSS reflects this on the results in an interactive and responsive manner regardless of the data size or scale.

Moreover, at any state of the search session, since the search result set will never be empty, the FSS continuously aggregates, groups and organizes the searched objects in a meaningful and concise way. This improves the usability of the system and gives the searcher the ability to learn and understand the information space being searched [2, 13, 23, 31, 32] and others.

2.1.3 Information Needs

From user information needs perspective, FSS can be classified into two classes: precision-oriented systems and recall-oriented systems [3].

Precision-oriented Systems

In precision-oriented systems, users look for one target resource. For example, in e-commerce systems the user intention is to locate a specific product to buy. The user intent might be meta-information about a single or specific resource, like the location for a specific store. In these systems, the FSS supports the user by presenting only relevant results and helps the user narrow down the information space quickly without getting lost in the system. User Interface (UI) used in this category should guide the user by presenting the relevant search results in a concise and focused manner.

The search task in this case is usually well-defined, users know what they are looking for and can recognize the sought resources and their associated facets as soon as they see them. FSS in this category aims at finding methods to minimize user effort and time-spent in allocating the desired resource.

As mentioned earlier, precision-oriented systems are widely adopted in e-commerce domain, examples are [33, 34]. Technical support is another domain where FSS was developed to help support personnel locating a specific troubleshooting document through facets, in order to help their customer solve the complaints [35, 36].

Recall-oriented Systems

On the other hand, recall-oriented faceted search systems are exploratory in nature. They aim at educating users about the information space being searched, where the users are interested in locating a group or set of resources rather than a single resource (like in precision-oriented systems). Users of recall-oriented FSSs carry out educational or investigative search tasks.

The FSS which belong to this category implements advanced visualization, aggregation and summarization techniques to support the complicated user interaction model. Aggregation and summarization techniques help the searcher gain insights about the content and its organization. They are also needed to support the user navigation by going back and forth to explore sub-spaces of the data being searched in a responsive and flexible manner.

Search tasks in recall-oriented FSS are open-ended, iterative, incremental, they target multiple items or attributes, and they also involve uncertainty since users do not know what they are looking for, and they are gaining knowledge as they browse the data.

Examples for recall-oriented systems are [4, 32]. A system called Hippalus [29] experimented with a search task in politics domain, the users were asked to educate themselves about the political parties participating in the elections. This task had no defined final target and each user could take a different path to achieve it.

Digital libraries are another domain where most of the search tasks are recall-oriented. Researchers use FSS to learn about new research areas and directions [8, 37, 38].

Identifying the type of information need in the search task has a major impact on the FSS. The information need governs which parts of the information space will be presented to the user and in which order. Will the FSS aggregate and summarize all data and facilitate exploration for recall-oriented search? Or, is it going to locate and focus only on relevant portions of the information space to present it to the user in a precision-oriented search?

This decision also affects how the search system is going to be evaluated, in recall-oriented systems the user spending more time and interacting more, can actually be a good sign. On the contrary, in precision-oriented systems this can be a sign of a poorly designed FSS, as these systems aim at helping users to finish their task as fast as possible with minimum effort.

In our research, we focus on precision-oriented FSS. Such systems target minimizing user effort to find their desired resource, hence, finding and ranking relevant facets to be presented to the user will reduce the effort needed to find the intended search target.

2.1.4 Underlying Data Structure

The data structures used by FSS can be categorized into three main categories:

- *Structured data*: The underlying dataset is derived from well structured knowledge graphs or linked data. Resources in this case are entities, and facets and their values are collected from the entities' types, ontologies, attributes or properties. Faceted browsing is the de facto standard for navigating structured datasets [3].

  However, the faceted browsers based on knowledge bases still struggle when dealing with large volumes of triples. These methods require extensive querying of the triple stores to collect data about the facets and their values, in order to support dynamic and interactive interfaces. Therefore, most of the systems using this approach are evaluated on small, domain specific ontologies [3, 12, 29, 32].

  Several software engineering and architectural considerations are involved to decide how the data will be stored and retrieved in RDF stores in an interactive and responsive manner. In some cases, tools like Facetize are developed to prepare and transform structured data for faceted search [39]. Examples of FSS operating on this kind of datasets are: [12, 25, 26, 29, 32].

- *Unstructured data*: In this kind of datasets, the resources being searched contain unstructured data for example: audio, images, or text like web pages or user tweets. However they still share some common characteristics, which can be deemed as facets. Special techniques are used according to the data type and search task, to extract facet values from the unstructured data. Design aspects related to processing time and complexity for extracting this kind of data need to be taken in consideration when building these systems. Example systems are: [6, 7, 17, 40].

- *Semi-structured data*: Where resources are objects that have some structured attributes or metadata, but they are also associated with unstructured data like

long textual research papers, or images or audio files. The majority of facets in these systems are obtained from the structured part of the data (i.e. attributes and their values). However, in some cases they can also be extracted or generated from the unstructured part, like for example top keywords in research paper for an academic search engine. Example systems are: [2, 14, 18, 41, 42, 43].

FSS underlying information structure determines how the facets are extracted. In structured datasets, it is a straightforward process since the facets and their values are directly collected from the properties of the resources. The FSS is then responsible for filtering, aggregating, organizing and ranking the facets before presenting them to the searcher.

An additional step is added in FSS functioning on semi-structured and unstructured datasets to generate facets or their values. Several algorithms are employed for this mission depending on the search domain and search task. Considerations related to how the extracted filters will be applied to navigate the data are also important when designing these systems. They also include the risk of propagating errors from the facet generation phase to the following phases of FSS.

Our proposed approach functions over semi-structured datasets, where facets and their values are extracted from the structured section of the data. However, the unstructured part is utilized to generate useful features for facet ranking.

2.2 Facets

2.2.1 What is a Facet?

The word facet means 'little face', it is used to describe objects which have multiple sides. The term originated from the Facet Theory and extended to be used in information science [24]. Recently, the term has been perceived as the aspects or dimensions which describe an item or information object. Multiple independent facets, in faceted search context, provide alternative ways of getting to the same item [24].

Facets can be seen as conditional filters that facilitate browsing the information space. Users select facets to "zoom in" or narrow the search results, or they can deselect facets to "zoom out", in this way widening the search scope. They can also move from one set of resources to another, using multiple navigation routes.

A recent user study aimed at understanding faceted search from human perspective [13], concluded that users interact with facets from the beginning to the end of the search session. In their experiments, the authors found that searchers employ facets distinctively at different stages of their search, and that users also use the facets implicitly without applying them on the search results. In this case facets support the searchers in learning and understanding the information space they explore. It was also mentioned in the study findings that although most participants like the faceted search, some of them were concerned about the choice overload introduced by facets. This shows how important it is to carefully select and rank the most relevant facets to the users.

In IR literature, the term *'facet'* is used to denote the criteria or the field which the filtering will be applied to in the resource. On the other hand, the term *'facet-value'* refers to the specific literal or entity used when deciding if the resource should be included in the result set or not. For example, in library domain, the book author, title, publication year are considered facets. The facet values in this case are: William Shakespeare, Romeo and Juliet, and 1595.

Niu et. al [13], argue that facets should not be confused with traditional search filters. While the authors acknowledge that they share some common characteristics, since they are both used to exclude items which do not satisfy a certain criteria, they are different, as facets cover several dimensions of the data. whereas search filters are simply applied to a single dimension.

In addition to that, facets extend the concept of filtering by covering complex data structures and hierarchies. Furthermore they aid the user in learning and understanding the information space being searched. They also educate the user about what is available and provide a means to reach and explore the data.

2.2.2 Different Facet Types

As facets deal with complex variety of data structures, several categorizations can be used to classify facets and their values. From UI design perspective, Vandic et al. [33] classified facets based on the data type of the facet-values they contain. According to their classifications, facets can be either *Qualitative* or *Numeric* facets. Example for numeric facets are age and price which contain only numerical values. Where qualitative facets are further classified into *Nominal* facets and *Boolean* facets. Boolean facets can have the value True, False, or unspecified, whereas nominal facets contain any number of literal values, like product display type, or movie director.

From another perspective, facets can also be categorized according to the structure of the facet-values belonging to this facet [9]. Facet-values can be flat, like author names or colors of t-shirts. They can also be hierarchical, for example, the facet country with value equal to 'Ireland', which belongs to 'Europe' in the countries taxonomy. Facet-values can also be grouped into ranges like product price range, or event dates grouped by year.

Tzitzikas et al. [3] categorized facets extracted from structured or semantic data into two main groups. In the first group, facets are extracted from `isA` or `isSubClassOf` relationships are called *Type-based Facets* (t-facets). They identify types of resources in the information space. The values in this case, can be flat but most commonly they belong to a multi-level taxonomy. In this case they are also called Hierarchical Facet Categories [44].

In the second group, facets which are collected from other entity attributes or relationships with other entities are called *property-based Facets* (p-facets). In contrary to the previous group, p-facets often have flat values, they can also belong to a hierarchical taxonomy but it is a less frequent case.

This categorization is applicable to the majority of FSS (see table 2.1). It can be adopted regardless of the underlying data structure used, i.e. beyond the semantic data representation. Structured data which involves resources with several classes can have t-facets driven from the types taxonomy. Some faceted browsers utilize only

t-facets, specially when they operate on resources with rich hierarchical taxonomies.

Hearst introduced a faceted search system which uses several t-facets to navigate food recipes [44]. The author suggested using multiple categorical t-facets rather than employing one large taxonomy. Other examples for systems using only t-facets are [5, 45, 46].

One key challenge for systems adopting t-facets is that the hierarchical taxonomy from which the t-facets are derived needs to be pre-defined. In case of structured data, this taxonomy can be generated from the class ontology. In other cases the taxonomies are manually defined by the owners of the FSS.

In the absence of an existing taxonomy general ones like WordNet are adopted in literature, this involves a mapping step from resources to their corresponding WordNet types [9, 23, 44, 47]. FacetX also attempts at overcoming the taxonomy limitation by automatically constructing the taxonomy from retrieved top results [46].

Other FSS use the two facet types but they handle them separately by showing the t-facets first, so the searcher can determine the type of resources first before looking into other p-facets [11, 25]. DFS [9] ranks the top t-facets first, then it selects the top p-facets for each t-facet to be presented to the end user. The idea of grouping and presenting t-facets hierarchy first before other p-facets, is widely followed by the e-commerce and shopping websites, where customers choose the department they are interested in first, then they use other attributes to filter the search results.

Other FSS which are based on a single resources type usually use only p-facets. Majority of the remaining FSS mix the two facet types and handle them in the same way [7, 17, 18, 48] and many others reported in the summary table 2.1 at the end of this chapter.

Understanding the characteristics of the facets in the system is crucial to the development of the FSS. It affects both the back-end design of how facets are retrieved, grouped, and ranked. It also dictates how the conditional filtering occurs on the data. On the other hand, from a front-end perspective, different organization and

visualization techniques can be chosen to present the facet according to their facet-values types and structure.

2.2.3 Facet Generation Methods

Facet generation is defined as the task of automatically discovering and extracting facets of information objects from their textual content [45]. The methods used in generating the facets and their values rely heavily on both the underlying data structure and the domain of the FSS. Several approaches have been adopted by faceted browsers in generating facets for text-based systems. In some domains, like e-commerce websites, the facets and their values are predefined by the admins or domain experts.

Other systems built for scientific publications also have static predefined facets (e.g. author, year, publication type, keywords, etc.), but their values are automatically extracted from the articles using Natural Language Processing (NLP) techniques [2, 8, 37].

Faceted browsers can also generate facets automatically from a document selection [14, 45] or Wikipedia text [7, 11, 25]. These approaches employ NLP algorithms, entity recognition, and sentence parsing to identify the facets and their possible values from the unstructured text. These approaches require expensive processing and indexing for the document set; they are usually very domain specific and hard to scale for large document collections.

Kong and Allan [6] made an attempt to extend the same facet generation concept to the general web. This is achieved by querying traditional search engines first to retrieve relevant documents; then, they apply the same NLP steps to generate the facets. Similar ideas allowing faceted exploration for search results were also introduced by Faflios et al. [49] and Kitsos et al. [50].

NLP based facet generation approaches are scalable to the general web. They also allow the user to digest and review large amounts of information in one page rather than opening many links in different pages, which can be a tedious and time

consuming task. However, approaches based on document collections still lack an understanding of the relationships between entities, and as the values of the facets are only those extracted from text, these approaches might also lose coverage of all possible facet-values.

A third mainstream approach in FSS, relies on knowledge bases. In this case, the data source for facets is represented by entities which are linked/connected through properties or attributes. Most of the developed FSS in this category are based on ontologies and RDF data [2, 3, 29, 30].

Facets generated from knowledge bases provide a better understanding of relationships between entities and their types than those generated from unstructured data. With the increased adoption of Linked Open Data (LOD) initiatives, knowledge bases can also compete with wider data coverage for facets and their values. Moreover, the extraction of facets based upon these well structured ontologies facilitates the process of organizing, grouping, and aggregating the facet-values by utilising relationships like `subClassOf` and `subPropertyOf`.

Abel et al. suggested a faceted search system for twitter data [17]. The system semantically enriches tweets with DBpedia using entity linking. The generated facets are then the types of entities tagged in the tweet and the facet-values are the extracted entities.

The idea of linking unstructured text to knowledge bases to enable faceted browsing of the textual items was also adopted by Inan et al. [7]. They developed a system to extract entities, which therefore enabled browsing construction health reports using faceted search means.

In order to avoid error propagation from the facet generation phase to the facet ranking one, our proposed approach uses t-facets directly derived from the dataset ontology. Since no generation step is needed, the evaluation will solely reflect the ranking step performance.

2.3 Facet Ranking

Despite the importance of this problem, it is rare to find studies dedicated solely to the investigation of facet ranking methods, nor to how this step, aside from other aspects of faceted search, affects the user's search experience. It is usually considered a complementary step to the facet generation process. The following ranking methods are collected from literature related to several faceted search systems.

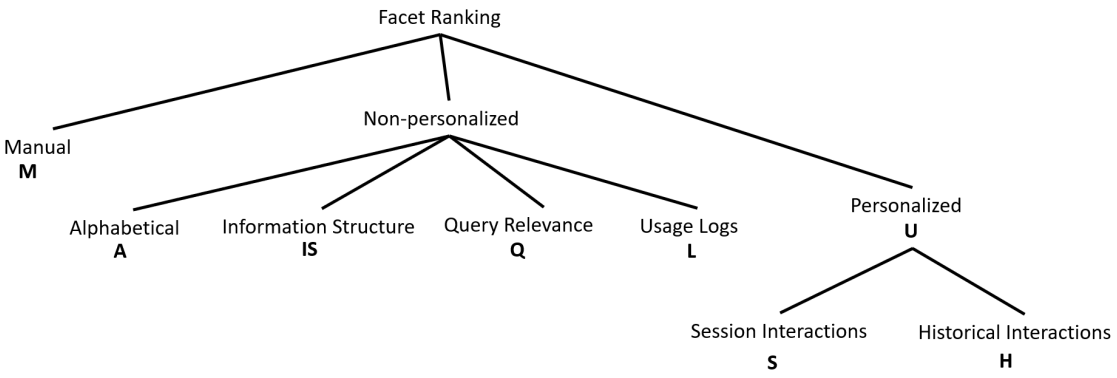


Figure 2.2: Summary of Facet Ranking Strategies.

Figure 2.2 shows a summary of the strategies followed in ranking facets in general. The figure is an extended version of the one proposed by Tzitzikas et al. [3]. In this version, the personalization strategies are included. FSS literature uses single or combined strategies to select the top facets according to their use case or search task. In the following sub-sections, examples for each approach are included.

2.3.1 Manual Systems

Generally speaking, many of the established faceted search applications use a manual facet selection process, which provides a static list of pre-selected facets to the users. The list is usually determined by domain experts [8, 10]. In addition, they might provide ranking methods for the facet-values. Other systems start with a manually defined list, however, giving the users the option to re-arrange the facets according to their preferences to support their exploration needs [29, 51].

As the magnitude and dimension of the data increase, this manual selection process becomes impractical and subject to human bias. Moreover, the importance and order of facets can change during the search session and change from one person to another. Predefined facets may not be helpful to the information seekers [13].

2.3.2 Non-personalized Methods

Information Structure Based Ranking

Facet ranking based on the information structure of the underlying collection is a very popular family of methods in faceted search literature. In order to induce facet importance, they leverage the structural characteristics of the facets, their values, or the items they cover. One of the most well known systems, Flamenco, sorts the facets and facet-values alphabetically [23, 44]. Arranging facets and facet-values by frequencies in the dataset is adopted by several systems including VisiNav, Faceted Wikipedia and many others [25, 27].

Systems that operate using predefined facets, also use predefined facet ordering. They rank the facet-values according to their frequencies in the dataset [2, 37]. Such ranking methods neither reflect the general importance of the facets nor the relevance to the user's interests.

Oren et al. [12] ordered the facets alphabetically; however, they used automated ranking to highlight the important facets. Facets are highlighted by changing the scale of their font-size in the UI. The proposed ranking method is based on the information structure of data. It computes a *navigation quality* score using weighted multiplication of several facet metrics. The metrics include predicate frequency, objects' cardinality, which indicates how many items belong to this predicate, and finally predicate balance, which is a score that reflects to what extent this predicate balances the navigation tree.

SemFacet [30, 32, 52] introduced a heuristic score for facet ranking. The score combines three characteristics: 1) diversity, 2) selectivity, and 3) nesting depth. The first characteristic favors facets which cover new items that are not covered by other

facets. The selectivity aspect prefers facets that will narrow down the search space more rapidly than other facets. The third characteristic, nesting depth, prefers facets with higher depth values in the facet hierarchy. Finally, the three characteristics are combined using multiplication. The approach is based on information structure and does not take the query nor the user into account.

The idea of minimizng navigation cost was also followed by FACeTOR [53], they rank facets based on heuristics which aim at minimizing navigation cost. The heuristic approximates a weighted version of the known SetCover method. SetCover heuristic is originally introduced by [47], it counts how many unique items are covered by this t-facet value. In order to obtain this count, a greedy approach is followed. It selects facets with highest count, then marks all the facet's associated items as covered, then it selects next facet as the one with the highest not covered item count and so on.

Another information structure-based ranking method was proposed by Feddoul et al. [54]. The method takes into consideration intra-facet and inter-facet metrics. Intra-facet metrics assign scores to each facet independently from other facets. The score favors popular facets, those with less facet-values, and facets with similar number of facet-values. On the other hand, inter-facet metrics focus on the relationships between different facets. For that, they calculate a score that favors facets which are not similar to each other, this will help diversify the final facet list. Similarity between two facets is derived from the shortest-path and the depth between them in the knowledge graph. The method starts by calculating the intra-facet metrics, then it proceeds with sorting the top N facets using the intra-facet metrics, then it follows a greedy approach in selecting the facets with the best inter-facet score.

Commercial systems started by manually pre-defining the order of facets. However, with the increasing amount of information available, they moved toward automatic selection and ranking methods. Faceted product shopping systems are usually precision-oriented, they target minimizing the effort needed by the customers to allocate the desired product.

Faccy [34] employed a selection algorithm which aims at partitioning the search

space in the most effective manner. The algorithm enables the user to finish the search task with the least amount of drill-down.

The method calculates facet utility given the top-m products retrieved by the search engine after a query is being submitted by the customer. Several models were introduced to calculate this utility score. They combine the probability that this facet contains the target product and the expected drill-down effort needed to reach the same product. Finally, facets and their values are sorted according to their best utility score.

Another system for product search, suggested ordering facet hierarchies in product navigation that minimizes the drill-down needed to reach a product at any stage [33]. The approach ranks facet-values based on the entropy of the products they cover and how they best split the navigation tree; the calculated entropy is then weighted by the product count. This approach follows the steps of an older system called Facetedpedia. It also optimizes a navigational cost model to enhance the facet hierarchy ordering [11].

Query Relevance Based

Ranking methods in this category consider the submitted query and/or the retrieved resources in response to that query. Facetedpedia [11] also introduced an algorithm to select and rank the p-facets hierarchies given the input query and the retrieved set of target articles by a keyword-search engine (document level search engine). The algorithm employs several navigation cost metrics based on the values of the p-facets and the number of the links in the target articles. The authors also developed a separate model to generate and order the Relevant Category Hierarchy (RCH), which utilizes the input query, the retrieved articles, and the cost calculated in the previous step. They refer to t-facets as RCH-Induced Facets. In order to find the top-N t-facets, they employ a hierarchical t-facet taxonomy and the p-facets calculated cost to choose the sub-graph of t-facets with minimum cost. At the end, the t-facets are presented to the information seeker as a plain list of categories.

In a similar way, query relevance is also considered by a system called IOS [49]. The authors proposed a ranking method using frequencies of t-facets weighted by their rank in the top results. Therefore, a facet appearing in the lower results will receive a lower score than those at the top. The concept of deriving facet relevance from the relevance of the documents it covers, is chased by Glass et al. [55]. After a traditional IR system ranks the set of relevant documents according to the input query, the approach calculates the facet score using a greedy method which maximizes the facet Discounted Cumulative Gain (DCG) value. The facet DCG score is obtained by aggregating the inverse of the ranks of the documents associated with this facet.

Liberman and Lempel [56] modeled faceted search session as a query followed by a single drill down. They combined information structure with query relevance, to rank facet-values. The ranking finds the top facet-values which will promote the target document to the first page of results. Query relevance is achieved in two ways: firstly, only facet-values in the returned search results will be considered in ranking; second, the score of the document returned by the search engine is utilized as an indicator of whether this facet-value covers more relevant results or not.

AFGF system links cluster of facet terms to their corresponding WordNet taxonomy using the Jacquard distance between the terms and the input query; the similarity score is then used to prune the WordNet tree and select a facet list to be presented as main topics. [45].

DFS [9, 36] introduced an approach which produces categorical facet ranks based on the facet's relevance to the input query. Then, the top categorical facets (t-facets) as well as their most common five p-facets are collected and presented to the user. The approach establishes relevance between t-facet and query based on their similarity in the vector space using K-Nearest Neighbour (KNN) algorithm. The vector space representation is generated using searched objects' entity embeddings.

PreFace [5, 57] is a FSS which retrieves concepts' prerequisites in form of t-facets. First, it calculates a score which reflects facet and query relevance. The score considers two aspects: 1) the similarity between the query language model and facet language

model, 2) the quality of the facet depicted by how many entities belong to each facet and the similarity between those entities, as it favors t-facets with higher count of entities and stronger inter-entity similarity.

The approach balances the trade-off between query relevance and concepts diversity. To achieve that, PreFace follows a greedy approach which first selects the best facets according to their relevance score. Then it iterates over the remaining t-facets to pick the ones with higher diversity from the already selected facets.

Usage Logs Based Ranking

Ranking the facets according to statistical analysis of search query logs is adopted by Zwol et al. [58], they use machine learning to combine signals from several information sources to train that model. In addition to that, they use user clicks as a feedback to build a ground truth for the proposed learning model [59, 60].

Other shopping systems have started to pre-compute facet ranks from user query and click logs [3]. This approach might reflect the importance of facets from the customer's perspective. However, it assumes the availability of a large amount of historical logs to derive these kinds of statistics.

The discussed facet ranking methods so far either rely on the structure of the underlying dataset, or presume that facet relevance is associated only with query relevance, or that facet importance can be inferred from users' collective usage logs. Non-personalized facet ranking methods neglect the individual user preferences. These methods might reflect a degree of relevance but the relevance of the facets can be both user and situation dependant, which is not addressed by any of these approaches.

2.3.3 Personalized Methods

Although the state of the art approaches on faceted browsing presents a wealth of research in this area, few approaches have analysed the role of personalization in faceted search, and how adapting the facet ranking according to users' preferences can impact their search experience. Personalization is based on individual user profiles collected

from their explicit interests, or inferred from their current or previous interactions in the system. Personalization usually happens at the facet ranking phase and/or the document retrieval phase. This section focuses on FSS personalization methods from a facet ranking perspective.

Session Based Ranking

Factic [15, 16] is an FSS that personalizes by building models from semantic usage logs. Semantic usage logs store the triples visited by the users and the timestamp indicating when they visited those triples. Several layers of user adaption are implemented and integrated with different weights to enhance the facet relevance model. The ranking model contains in-session user behavior (highest weight), combined with the aggregated behavior for the same user collected from their previous sessions. The model also includes information from similar user profiles and global user statistics (lowest weight).

However, this model suffers from the cold start problem. It requires a considerable amount of users to provide a new user with meaningful ordering for facets. While the ordering might reflect the general importance of the facets and their values, it still misses the opportunity to reflect the individual user interests. The model takes time to implicitly infer user preferences. Users might be bored and leave the system by then.

Interestingly, during their evaluation, the authors found that suggesting suitable ontological concepts from user's history improved the total task completion time and decreased the number of user clicks. This is due to the fact that these hierarchical concepts created effective navigation shortcuts.

Sah and Wade [4] also employ session-based user interaction to personalize search concepts (t-facets). As soon as the user selects a t-facet, the system re-organizes the others according to their similarity to this concept. It also personalizes (re-ranks) the retrieved documents according to the selected concepts and query expansion. The proposed system did not address the hierarchical nature of the categories.

History Based Ranking

Koren et al. [14] suggested a collaborative and a personalized ranking by leveraging explicit user feedback about the facets. They take user ratings into consideration to build a relevance model for individuals. They also use the aggregated ratings to build a collaborative model for the new users in order to face the cold start problem and provide initial good facets in absence of a user profile.

An approach to personalize facets by matching them to a user profile collected from social media was proposed by Le et al. [31]. It builds user profiles from their social media profiles. It collects objects with which the user interacted and uses them to infer their interests; then it maps those interests to Wikipedia articles to build the user profile. This is achieved by extracting top TF.IDF terms from the user favored social media objects as well as the articles associated with facet. While displaying the facets, the algorithm highlights the facets according to the number of matched terms with the user profile. The same idea was followed by Nguyen et al. [40]. He et al. [61] adopted TF-IDF score to rank the top categorical facets to the user, the categorical facets were extracted from user previous emails to provide interactive visual facets to the user.

Adaptive Twitter search system implements four strategies to rank facet-values [17]. The first strategy is based on facet-value frequency in tweets. The second strategy personalizes the ranking based on a user model. The user model contains entities extracted from the user's old tweets. The facet-values are weighted higher if they exist in the user profile. The third strategy favors facet-values from more recent tweets. The fourth and final strategy uses diversification to order the facet-values. To achieve this, it ranks the facet list using the first strategy, then it randomly picks facet-values from the end of the list and assigns higher ranks to them.

Bivens et al. [35] proposed a personalized facet ranking method using query relevance and user profiles. User facet profiles are collected from previous search logs, where the query relevance is established using topic models. The query is used first to retrieve a set of relevant document results, then facets are collected from these

results. The facets are ordered based on users' facet profiles collected from their recorded usage logs.

Our approach solves the FSS personalization problem by using user profiles collected from their historical ratings in the system. The approach is concerned with type-based facets and how they can be ranked while preserving their multi-level hierarchical structure. This area is not covered by earlier literature in facet ranking research.

2.4 Evaluating Faceted Search Systems

It is crucial to understand how existing work evaluated their FSSs. The final target of the evaluation depends on the search task and the information need tackled by the system. Even within the same search task, existing FSSs vary on the strategies they follow to measure the success of their proposed systems.

Surveying published work related to faceted search evaluation in general, and facet ranking evaluation in particular, was essential in articulating the evaluation strategy followed in this thesis. FSS evaluation strategies can be broadly divided into two types: Task-based evaluation using real users or simulated user evaluation. This survey covers both strategies in the following sections:

2.4.1 Task-based Evaluation

In task-based approaches, researchers recruit a number of participants to experiment with the developed FSS. Participants are given a search task to fulfill using the system. Researchers on the other hand employ different methods and metrics to asses the effectiveness of their systems. Robert et al. [62] evaluated their FSS using pre- and post-task questionnaires to measure the participant's knowledge level and satisfaction with the system.

A similar approach was followed by Hippalus [29], in which the system developers prepared a user study with 38 people and asked them to watch a video tutorial, then

complete a search task, and then answer a questionnaire. Both studies measure user satisfaction based on the questionnaire answers and used this as a metric to assess the system success.

This type of evaluation also utilises exploratory search evaluation metrics, in case they are recall-oriented systems. In this scenario, the user interaction is observed to compare different facet ordering or facet generation techniques. Measurements are collected from search logs, click behavior and eye tracking tools. In addition, metrics like time to complete the task, count of query reformulation and task success are calculated. All these metrics are then analysed to study user behavior in the system [3, 12, 16, 23, 29, 33, 61, 63].

While task-based evaluation is popular among recall-oriented systems, it is also adopted by precision-oriented systems. In this case, the same metrics can have different objectives, for example precision-oriented systems aim at minimizing user clicks and task completion time, while recall-oriented systems might favor more interactions from the user and longer session times.

Factic [15], which is a precision-oriented system, followed a task-based evaluation strategy. They asked users to implement specific scenarios (like find jobs with specific properties) and recorded the number of clicks and the task completion time.

2.4.2 Simulation-based Evaluation

Koren et al. argued that user based studies, while undoubtedly useful, are very limited, because they are expensive to conduct, hard to repeat, and the number of users is usually limited, which makes their results inconclusive and not reproducible, especially in personalized search systems. They instead suggest an approach that simulates the clicking behavior of users in the context of FSS [14].

The proposed evaluation approach attempts to measure how well user information need is satisfied compared to the amount of effort required by a user. User information need is assumed to be fulfilled by locating the target resource using the FSS. Based

on this idea, the proposed evaluation rewards users actions taken towards finding this intended target. The goal of the evaluation is to minimize the effort needed by the user to fulfill his/her search needs.

This is the most adopted simulation model for precision-oriented faceted search systems in literature [1, 17, 33, 34]. For example, Adaptive Twitter search system [17] followed this evaluation approach to simulate users finding tweets. They use this evaluation method to compare non-personalized and personalized facet ranking techniques.

Vandic et al. [33] also adopted a simulated user evaluation method. Different models for clicking behavior are used and metrics related to user effort to scan facets and their values are computed, that is, how many times a facet or facet-value are (de)selected. They also calculated some performance measures like computation time needed to retrieve ordering of facets, and session success, which is defined as the number of clicks needed to find the target resource.

INEX 2011 Data Centric - Faceted Search task [1] also adopted the same user simulation model. Moreover, the task organizers developed a number of evaluation metrics to measure the user effort needed to find the first relevant target. They also introduced other metrics to measure how many relevant results are covered by a given ranked facet list.

This research follows the steps of INEX 2011 Data Centric Faceted Search task evaluation methods. This provides a well established evaluation strategy suitable to answer our research question. Chapter 4 details the simulated user interaction model and the used metrics.

2.4.3 Other Methods

FSS researchers also report a variety of other metrics when evaluating their systems. For instance, Facet Embeddings [37] studies the quality of the generated facet topics in comparison to standard NLP topic-generation methods. Dakka et al. [47] also focus

on evaluating the quality of the generated facets and their ranking, they defined two metrics: 1) coverage, which measures the extent to which the generated facet hierarchy covers all data (i.e. all results are reachable using the generated hierarchy), the higher the coverage the better the system, and 2) cost, which is the average path length to reach any object from the root of the tree, the lower the cost the better the system.

Faceted browsers also include additional performance evaluation measures, especially when the system operates on large scale or complex data, or when the system needs to do complicated NLP processing to extract facets or their values and aggregate large volumes of data [17, 22, 52].

Performance metrics include: run-time and responsiveness of the UI from the query submission till the results page is populated, query execution time, and time to update the interface after a facet is selected or deselected. In addition to that, it is common practice to report the machine specs used in running the experiment. These metrics are compared against other systems on different datasets with different scales (for example: number of triples).

2.4.4 Evaluation Domains & Collections

Literature in faceted browsing experimented with a wide variety of domains and search tasks. Some systems were developed for certain search verticals. For example, faceted search is the dominant approach used in e-commerce websites; therefore, several authors use product search as a domain to evaluate their faceted search systems [33]. Flamenco and Zwol et al. both used digital image search as the domain for their experiments [23, 58]. He et al. [61] evaluated their system in the email search domain, where Glass et al. [55] experimented with two datasets in the technical question and answer systems.

Digital libraries and scientific publications are another domain that received a great deal of attention in faceted search literature [2, 16, 37, 62]. This is because the search tasks in this domain are exploratory in nature, they involve learning and

investigating, and are open-ended. The data is also semi-structured and has clear common facets that describe research papers.

Other faceted browsers were developed based on specific data format (like semantic data) and they were customized and evaluated using several domains. Factic [16] evaluated their system by experimenting on multiple domains: scientific publications, job offering and digital images. Hippalus [29] was also evaluated in multiple domains like politics, sports and marine species .

Chantamunee et al. [42] suggested a personalized facet ranking based on Collaborative Filtering (CF). The MoviesLens dataset was used in their evaluation. The average rating given by the user to the facet is used as ground truth, they reported RMSE values to measure the effectiveness of the ranking method. This experimental setup might be useful in prediction tasks, but it does not assess how the final facet list will assist the user in reaching their target. The MovieLens dataset is not suitable for this work's task as movies genre (types) are limited and do not have a multilevel hierarchical taxonomy.

Due to the novelty of the personalized t-facet ranking task, this area lacks the existence of ground-truth datasets to be utilized during the evaluation process. Most of the reviewed literature in the facet ranking area use their own created datasets. The area misses a standardized process to create bench-marked datasets which can be used to compare different ranking methods.

For this reason, a set of criteria is established in this thesis to guide the selection of appropriate domain, search task, and dataset. These criteria are outlined in chapter 4, section 4.1.1. The thesis also proposes a ground-truth creation framework which customizes and transforms existing datasets to fit the purpose of the evaluation discussed in chapter 4, section 4.1.

The selected search task in this work has a clear context and is well defined. The information need type is precision-oriented. The tourism domain is chosen in this

thesis' experiments. The choice is motivated by the introduced criteria. Furthermore, there are a number of published datasets that can be used as a benchmark for evaluation. In addition to that, it has clear potential for personalization to improve the delivered search experience. For example, in the tourism domain the search task could be "Find one interesting activity to do this weekend in New York" rather than "Find interesting activities in New York": the first is more defined and sets a stopping point for the users.

2.5 Summary of FSS Classification

This section provides a summary table 2.1 which categorizes the existing literature using the aspects covered in this chapter. The first column contains system name (if available), otherwise it contains the authors' name. The publication year of the cited paper is presented in the second column. Table entries are sorted using the year column. The third column contains the information need (IN) categorization. It shows whether the system is Recall-oriented (R) or precision-oriented (P).

The classification of FSS based on the underlying data structure is outlined in the forth column (DS), with (Yes) meaning the data is fully structured, (No) meaning the data is unstructured, and finally (Semi) denoting semi-structured data. The domain(s) of evaluation of each system are listed in the fifth column, followed by the evaluation (Eval) column which states the adopted strategy to evaluate the FSS. The values for this column are: (T) denoting task-based evaluation, (S) marking simulation-based evaluation and (-) for unknown (not mentioned in the paper) or (O) for other methods.

The rest of the table columns focus on the facets in each FSS. Types of facets used by the FSS are shown in the seventh column, with values: (TF) meaning that the FSS operates only on type-based facets, where (PF) means the system operates only on property-based facets, and finally (F) means the system uses both types of facets. The next column summarizes the facet ranking approach (see figure 2.2 for notations).

The ninth column (Handling) highlights if the FSS uses the same ranking or display method for both facet types (p-facets and t-facets) or not. The value is only provided for systems which operate using both types of facets. The last column summarizes the used facet generation method: in structured systems, facets are usually derived from objects' attributes (Attr.), whereas in other data types, they might be generated manually (M), using NLP, Entity linking (EL) or a combination of them (denoted by the plus sign). The last row of the table contains a summary of the notations.

Method Name	Year	IN	DS	Domain	Eval	Facets			
						Types	Ranking	Handling	Generation
Flamenco [23]	2003	R	Semi	Images	T	F	A	Same	-
OntoViews [10]	2004	R	Yes	Museums	-	F	M	Diff	M
Dakka et al. [47]	2005	-	Semi	Images+TV+Web	O	TF	IS	-	NLP
Faceted Categories[44]	2006	R	No	Food	T	TF	A	-	-
MSpace [28, 51]	2006	R	Yes	Music	S	F	M+U	Same	Attr.
BrowseRDF [12]	2006	P	Yes	DL+CR	T	F	IS	Same	Attr.
Koren et al.[14]	2008	P	Semi	Movies	S	F	CF+U	Same	Attr.
AFGF [45]	2010	P	No	Medical+DL	O	TF	IS+C+Q	-	NLP
Facetedpedia [11]	2010	R	Yes	Wikipedia	T	F	IS+Q	Diff.	Attr.
Zowl et al. [60]	2010	P	No	Images	S	F	L	Same	NLP+EL

- IN : Information Need
  - R : Recall-oriented.
  - P : Precision-oriented.
- Eval: Evaluation
  - S : Simulation-based.
  - T : Task-based.
  - O : Other.
- Facet Types Handling
  - Same: Same.
  - Diff.: Different.
- Facet Types
  - TF: T-facets.
  - PF: P-facets.
  - F: T-facets & P-facets.
- Domain
  - DL: Digital libraries.
  - Web: General web.
  - CR: Criminal records.
- Facet Ranking
  - A: Alphabetical.
  - M: Manual.
  - CF: Collaborative Filtering.
  - Q: Query relevance.
  - C: Content based.
  - IS: Information Structure.
  - U: User based (Personalized).
  - L: Usage Logs.
- DS : Data Structure
  - Yes : Structured.
  - No : Unstructured.
  - Semi : Semi-structured.
- Facet Generation
  - M: Manual.
  - Attr.:From Objects Attributes.
  - NLP: Using NLP Techniques.
  - EL: Entity Linking.

Table 2.0 Summary of Faceted Search Systems in Literature – continued on next page

Method Name	Year	IN	DS	Domain	Eval	Facets			
						Types	Ranking	Handling	Generation
Faceted Wikipedia [25]	2010	R	Yes	Wikipedia	-	F	IS	Same	Attr.
Factic [15, 16]	2010	P	Yes	DL+Jobs+Images	T	F	U+L	Same	Attr.
FACeTOR [53]	2010	P	Semi	Cars+Movies	T	F	IS	Same	Attr.
AdaptiveTwitter [17]	2011	P	No	Social Media	S	F	U	Same	NLP
Let et al. [31]	2012	R	Yes	Web	-	F	U	Same	Attr.
IOS [49]	2012	R	No	DL+Fishery	T	TF	IS	-	EL
Faccy [34]	2013	P	Semi	E-commerce	S	F	IS+Q	Same	Attr.
Sah & Wade [4]	2013	R	Yes	Tourism	-	TF	U+C	-	Attr.
Liberman & Lempel [56]	2014	P	Semi	Web	S	TF	L	-	M
FWS [6]	2014	-	No	Web	S	TF	C+Q	-	NLP

- IN : Information Need
  - R : Recall-oriented.
  - P : Precision-oriented.
- Eval: Evaluation
  - S : Simulation-based.
  - T : Task-based.
  - O : Other.
- Facet Types Handling
  - Same: Same.
  - Diff.: Different.
- Facet Types
  - TF: T-facets.
  - PF: P-facets.
  - F: T-facets & P-facets.
- Domain
  - DL: Digital libraries.
  - Web: General web.
  - CR: Criminal records.
- Facet Ranking
  - A: Alphabetical.
  - M: Manual.
  - CF: Collaborative Filtering.
  - Q: Query relevance.
  - C: Content based.
  - IS: Information Structure.
  - U: User based (Personalized).
  - L: Usage Logs.
- DS : Data Structure
  - Yes : Structured.
  - No : Unstructured.
  - Semi : Semi-structured.
- Facet Generation
  - M: Manual.
  - Attr.:From Objects Attributes.
  - NLP: Using NLP Techniques.
  - EL: Entity Linking.

Table 2.0 Summary of Faceted Search Systems in Literature – continued on next page

Method Name	Year	IN	DS	Domain	Eval	Facets			
						Types	Ranking	Handling	Generation
FacetTree [62]	2014	R	Semi	DL	T	F	-	Same	M+NLP
FeRoSA [8]	2016	R	Semi	DL	T	F	M	Same	-
Hippalus [29]	2016	R	Yes	Politics+Sports+Marine	T	TF	M+U	Same	M
Vandic et al. [33]	2017	P	Semi	E-commerce	S	F	Q+IS	Same	Attr.
Facet Embeddings [37]	2017	R	No	DL	O	TF	C	-	NLP
SemFacet [32, 52]	2017	R	Yes	E-commerce	O	F	IS	Same	Attr.
SemanticScholar [2, 38]	2018	R	Yes	DL	-	F	IS	Same	Attr.
Bivens et al. [35]	2019	P	No	Tech. Support	-	F	Q+U	Same	-
Feddoul et al. [54]	2019	R	Yes	Wikidata	T	F	IS	Same	Attr.
NaLa-Searc [48]	2020	P	Yes	Medical	T	F	-	Same	Attr.

- IN : Information Need
  - R : Recall-oriented.
  - P : Precision-oriented.
- Eval: Evaluation
  - S : Simulation-based.
  - T : Task-based.
  - O : Other.
- Facet Types Handling
  - Same: Same.
  - Diff.: Different.
- Facet Types
  - TF: T-facets.
  - PF: P-facets.
  - F: T-facets & P-facets.
- Domain
  - DL: Digital libraries.
  - Web: General web.
  - CR: Criminal records.
- Facet Ranking
  - A: Alphabetical.
  - M: Manual.
  - CF: Collaborative Filtering.
  - Q: Query relevance.
  - C: Content based.
  - IS: Information Structure.
  - U: User based (Personalized).
  - L: Usage Logs.
- DS : Data Structure
  - Yes : Structured.
  - No : Unstructured.
  - Semi : Semi-structured.
- Facet Generation
  - M: Manual.
  - Attr.:From Objects Attributes.
  - NLP: Using NLP Techniques.
  - EL: Entity Linking.

Table 2.0 Summary of Faceted Search Systems in Literature – continued on next page

Method Name	Year	IN	DS	Domain	Eval	Facets			
						Types	Ranking	Handling	Generation
Chantamunee et al. [18]	2020	P	Semi	Movies	S	F	CF	Same	Attr.
DFS [9, 36]	2020	P	No	Tech. Support	S	F	Q	Diff.	NLP+EL
FacetX [46]	2020	P	No	Jobs+Food+Movies	-	TF	-	-	NLP
He et al. [61]	2021	R	No	Email Search	T	F	U+IS	Diff.	NLP
HSEarch [7]	2021	P	No	Medical	T	F	IS	Same	NLP+EL
PreFace [5, 57]	2021	R	No	DL	T	TF	Q+IS		NLP+EL
Glass et al. [55]	2021	P	No	Tech. QA	S	F	Q	Same	NLP
RelFacet [63]	2021	R	Yes	DBpedia	T	PF	IS	-	Attr.

- IN : Information Need
  - R : Recall-oriented.
  - P : Precision-oriented.
- Eval: Evaluation
  - S : Simulation-based.
  - T : Task-based.
  - O : Other.
- Facet Types Handling
  - Same: Same.
  - Diff.: Different.
- Facet Types
  - TF: T-facets.
  - PF: P-facets.
  - F: T-facets & P-facets.
- Domain
  - DL: Digital libraries.
  - Web: General web.
  - CR: Criminal records.
- Facet Ranking
  - A: Alphabetical.
  - M: Manual.
  - CF: Collaborative Filtering.
  - Q: Query relevance.
  - C: Content based.
  - IS: Information Structure.
  - U: User based (Personalized).
  - L: Usage Logs.
- DS : Data Structure
  - Yes : Structured.
  - No : Unstructured.
  - Semi : Semi-structured.
- Facet Generation
  - M: Manual.
  - Attr.:From Objects Attributes.
  - NLP: Using NLP Techniques.
  - EL: Entity Linking.

Table 2.1: Summary of Faceted Search Systems in Literature

2.6 Survey Conclusions

This chapter surveyed key publications and research trends related to faceted search systems. It presented the faceted browsers literature from different perspectives. The chosen perspectives are those that are closely related or affect how the facet ranking happens. Other components, while no doubtfully crucial to the FSS, do not affect the facet ranking process, therefore are not included in the survey.

The surveyed systems are classified in table 2.1 using the aspects discussed earlier in the chapter. From the classification summary we can observe that majority of FSS operate on knowledge graphs or structured data [2, 3, 4, 10, 12, 16, 25, 29, 31, 32, 48, 51, 54, 63], or the structured part of semi-structured data [8, 14, 23, 43, 47, 53, 56, 62]. In this case the facets are derived from the existing attributes. Faceted browsers which operate on unstructured textual data are less common, they employ NLP or Entity linking techniques to generate facets [5, 6, 7, 17, 35, 36, 44, 45, 46, 55, 60, 61]. Other unstructured data types like audio or video are not explored by FSS literature.

FSS which are recall-oriented, generally favor task-based evaluation, as the search tasks are usually open-ended and involve uncertainties, it is hard to assess FSS using simulated users as it doesn't reflect the human complex cognitive behavior [8, 11, 23, 29, 37, 44, 49, 54, 61, 62, 63]. In this case, the evaluation domains also are exploratory in nature to support such search scenarios.

On the other hand, precision-oriented systems are suitable for some specific search scenarios and domains. As they generally target quick task completion time, they mostly adopt simulated evaluations [14, 17, 34, 36, 55, 56, 60]. Personalized faceted search systems also favor simulation based approaches, as for personalization to be proven useful, requires large scale experiments [14, 17, 18]. This is especially true in the situations where personalization is based on historical interactions rather than current session interactions.

It is also noticeable that most developed FSSs, operate on single domain [4, 5, 8, 10, 14, 17, 18, 23, 28, 32, 33, 34, 35, 36, 37, 38, 44, 48, 60, 62], some experimented with

multiple domains [12, 16, 45, 47, 53], and few FSS attempted open domains search like the general web [6, 31, 56]. Although FSS has proven to be useful in vertical search systems, adapting and developing methods to extend it to general web is an evolving research direction with potential benefits.

2.6.1 Research Gap

The current FSS literature rarely studied the effect that the facet ranking process has had on the evaluation metrics, separate to the facet generation step and other aspects of FSS. Questions related to how errors propagate from the facet generation phase to the facet ranking phase also remain unanswered. This is a direction that needs further investigation.

Reviewed faceted search systems vary in how they use and rank the facets. A number of systems provide only t-facets to the searchers [4, 6, 29, 37, 44, 45, 46, 47, 56, 57]. However, these systems do not handle the hierarchical nature of the t-facets during the ranking. The hierarchical nature of the facets are also neglected in other FSS which mix both t-facets and p-facets.

Systems which mix the type of facets rank them in the same way [7, 8, 12, 14, 16, 17, 23, 28, 33, 34, 62]. Some differentiate the t-facets only in the UI by displaying them separately from other facet types. It is less common for systems which rank both types of facets to provide a separate ranking for each facet type [10, 11, 36]. Type-based facets carry useful categorical characteristics which should be leveraged by the ranking algorithm.

Ranking mixed facet types usually employs techniques based on information structure, search logs or query relevance. This ranking might indicate the general importance and the query relevance of the facets, but it does not take into account the individual user interests.

Moreover, the majority of the personalized FSS employ the same ranking as both t-facets and p-facets [14, 16, 17, 29, 31, 35, 51]. Providing a personalized method for

the special case of t-facets based on historical user feedback is an area which is new and not explored by earlier literature.

In this chapter we discussed FSS from different perspectives, trends in literature were presented and organized with a focus on facets and their ranking. This literature reviewing process helped shape the final search task, evaluation method and the ranking process. In the next chapter, the new personalized t-facet ranking approach is presented in detail.

Chapter 3

Personalized Type-based Facet Ranking

3.1 Articulating A New Approach

The approach presented in this thesis is dedicated to solve the facet ranking problem. It is suitable for type-based facets as it leverages their categorical characteristics and handles their hierarchical nature. In order to avoid error propagation from the facet generation step, the proposed method utilizes t-facets which are generated from structured parts of the data, although it can use the unstructured part to generate features useful for ranking.

In order to solve the personalized t-facet ranking problem, the proposed ranking approach exploits user historical feedback to find out which t-facets are interesting to the user. It also organizes the relevant t-facets in a hierarchy to make it readable to the end user. All the studies reviewed, either fail to handle the categorical nature or the hierarchical structure that characterise type-based facets. Moreover, existing t-facet ranking methods do not take users historical feedback into consideration during the ranking process.

Choosing the vertical, the search task, and the information need of the FSS users has consequences on how the facets should be ranked. Should the ranking favor facets which will give a broader view about the information space to the searchers? Or should it select facets which will promote relevant documents and make it easier for the users to find relevant documents? Other factors are also considered, like the data structure employed in the FSS as well as the scale of the data collections being used. Besides their impact on the ranking process, these factors also play a role in deciding which strategy should be followed to evaluate the FSS.

The proposed approach objective is to minimize user effort in precision-oriented search tasks. It is evaluated using simulation-based methods which are well established and widely used in faceted search literature.

The tourism domain, specifically the POI suggestion tasks are chosen for the search task. In addition to the the availability of several datasets online to be used in experimentation, POI suggestion is a well-known personalization task, it is proven in literature that the category of POI plays an important role in the POI suggestion process [64, 65, 66].

This research utilises evaluation methods established in the faceted search literature. These methods are detailed in the experimental setup in chapter 4. In addition to that, several facet ranking baselines, selected from literature, will be compared using the same facet generation approach. This will allow the evaluation of the ranking step, and how it affects evaluation metrics, in isolation from the facet generation step.

3.2 Approach Overview

The proposed personalized type-based facet ranking approach consists of two steps. The first is an individual type-based facet scoring step. It assigns a relevance score to each t-facet. The score combines signals from the user profile, the documents ranked according to the input query, and other collaborative filtering features. Applying the scoring method to all the t-facet levels adds extra processing. To avoid this, a

second step is needed to propagate the relevance score from the leaf-node t-facets and use it to rank the higher level of parent nodes. This second step constructs the final type-based facet tree to be displayed to the user. The inputs in this last step are the generated scores for each facet obtained from the first step and the original hierarchical taxonomy from which the t-facets are derived.

Later in this chapter the two steps are described in detail. For the scoring step, three techniques for generating the personalised scores are presented in section 3.3. While, in section 3.4, three strategies for the final t-facet tree construction are suggested. It is understood that the later step is impacted by the Human-Computer Interface (HCI); however, in this PhD we focus on the t-facet tree construction from the angle of purely improving the IR metrics, i.e. finding the best relevant sub-tree that will minimize the effort needed by the user to find the relevant result. In real FSS design, it is acknowledged that HCI factors should be taken into consideration while, for example, deciding the length of facet display and the number of levels to be displayed to the user. For the purpose of this study we consider these two parameters predefined or pre-configured in the system.

Before moving to the details of the scoring and tree construction steps, the remainder of this section will layout some background information to help understand how the approach operates. The next sub-section 3.2.1, explains the overall search scenario in which this approach fits. After that, sub-section 3.2.2 will layout the data format needed by this approach to function. The formal notations used in the rest of the chapter are summarized in sub-section 3.2.4.

3.2.1 The Search Scenario

When the user submits a search query, the underlying search engine starts by ranking the original documents [1]. It retrieves the top relevant documents to the FSS. The FSS then collects the t-facets associated with all retrieved documents. The set of collected t-facets are then considered relevant and they are input to the t-facet ranking

[1]How the document ranking is performed is outside the scope of this research. However, the impact of this step on the facet ranking process is discussed in chapter 5, section 5.4

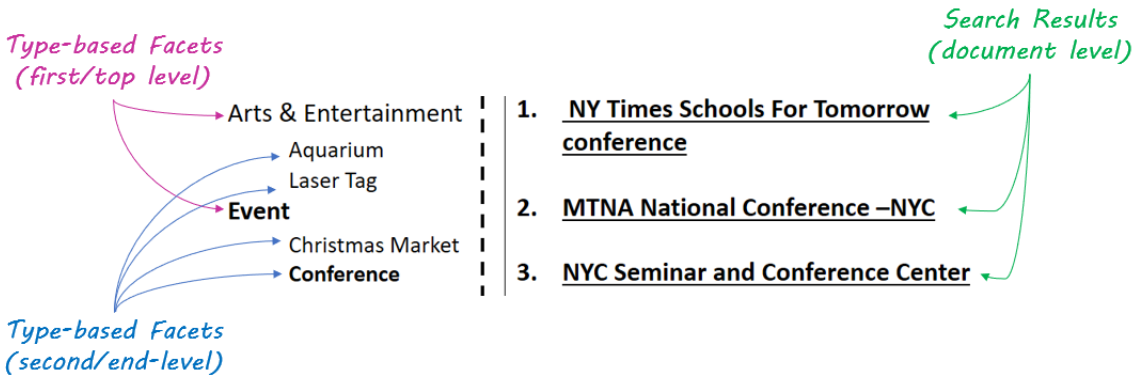


Figure 3.1: Example illustrating the multilevel ranking approach

approach.

Then, the t-facet scoring process uses a set of signals to produce the relevance score for each leaf-node or end-level t-facet. The signals are collected from the users previous ratings, aggregated document ranking scores, and general facet importance signals gathered from the data collection. After this scoring is returned, the tree construction step will employ this score to order the higher tree levels and decide how to group end-level facets. In other words, it rebuilds the final tree to be shown to the user.

The approach can also be seen as a multilevel ranking process. Figure 3.1 shows an example for this multilevel ranking process. The illustrated example operates on a two level taxonomy. During the search process, the ranking occurs first on the documents level (right side of the figure). The produced document rank, in addition to the user profile and other importance information are then used to rank the lowest level of type-based facets (t-facet scoring step). The tree construction step uses the scores of the previous level for ordering and grouping the higher level type-based facets (left of the figure). It repeats this step till all displayed levels are ranked and the final tree is generated. The user can select the t-facet to filter the document level results to only those associated with this t-facet, for example the Event and Conference t-facets highlighted in the figure.

3.2.2 Designated Data Structure

The intended dataset contains a collection of documents or resources to be searched. Each document has:

- Textual description, for example a web page content or a description written by the document's owner.
- A set of reviews, and ratings given to this document by the system's users, they reflect users experiences or opinions about this document.
- A set of categories assigned to the document by the document owner or system admin. The categories belong to a large hierarchical type taxonomy and they are treated as type-based facets. Documents are associated with end-level or leaf node types.
- Other properties related to the document are not included in this approach.

Every document in the dataset collection must be associated with at least one category. Documents can also belong to several categories at the same time. Each category must match with one and only one node of the hierarchical taxonomy of categories. The FSS system operates on a single taxonomy of unified types for all the collection. When the facet types belong to a large, multilevel taxonomy, the FSS needs to select the appropriate levels in the t-facet hierarchy to present to the user. In that case, we refer to them as level-n t-facets instead of facets and facet-values, where n is the level of the facet in the original taxonomy.

This hierarchical taxonomy can be seen as an directed acyclic graph or a tree of categories. Meaning that each node must have exactly one parent and can have zero or multiple children. Figure 3.2 demonstrates the tree structure from a leveling point of view. The taxonomy tree has a single root at level zero, this is the top of the tree. Level-n is the lowest level and it contains the end leaf nodes. The level of a type is defined by 1 + the number of connections between its node and the root. Categories or types at the same level lay at the same distance from the root node.

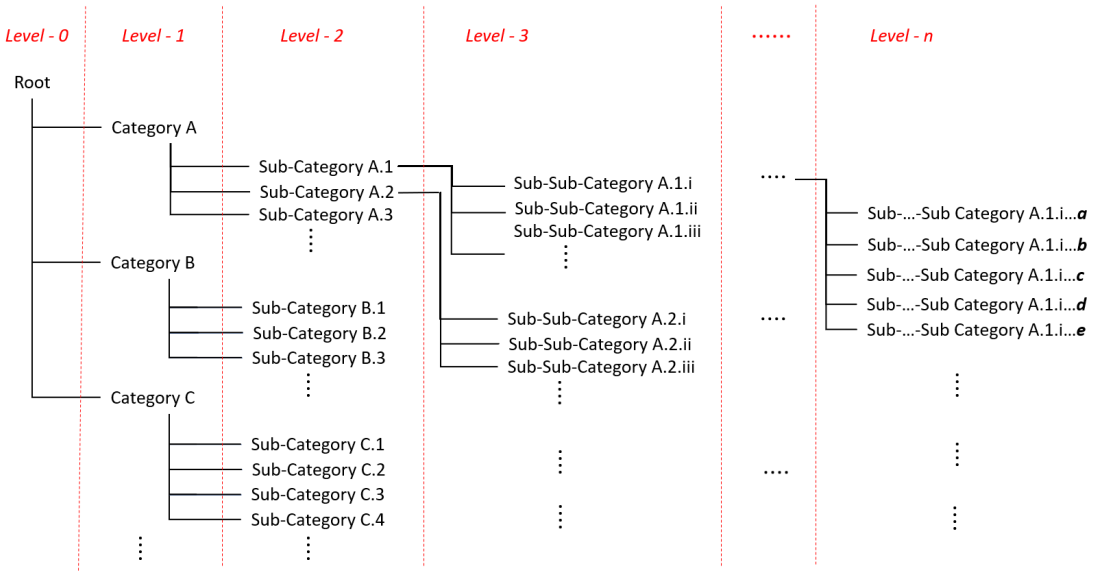


Figure 3.2: Demonstration of a multilevel hierarchical type taxonomy.

This categorical taxonomy serves as the type hierarchy from which all the type-based facets are derived. Defining this taxonomy, its levels, and the relationship between its nodes is crucial as it governs the t-facet ranking process.

Since this research is concerned with t-facet ranking rather than facet generation, the t-facets themselves are predefined. The t-facets are directly collected and aggregated from the data. How the t-facet taxonomy is created or assigned to documents is outside of the scope of this research. However, the ordering of t-facet-values is decided by the proposed algorithm. Having a predefined fixed t-facet taxonomy avoids propagating errors from the facet generation step to the facet ranking step. This enables the opportunity to assess and evaluate the ranking approach in isolation.

The indicated dataset also includes user profiles. Each profile contains basic information about the user like age and gender. It also includes historical ratings the user gave to a number of documents from the same collection. Users choose ratings for the documents they visited which reflects whether they favored those documents or not.

Rating values belong to a numerical scale where the minimum value means dissatisfaction, and the maximum value reflects complete satisfaction. This scale values can also be mapped, or classified, to *positive*, *negative*, or *neutral* labels. The higher half of the scale of sorted values is labeled as positive, the middle value is labeled neutral, while the negative label covers the lower values of the scale.

3.2.3 Required Pre-processing

A number of pre-processing steps are required for each document before or during the search engine indexing. The first and most important pre-processing step is performed on the categories defined for each document. It ensures that all document categories, as well as all their ancestors, are included in the original categories set of that document. If a missing ancestor is found, it appends it to the set of categories. If several categories are assigned to the same document, they may or may not have a common ancestor. In case they share a common ancestor, this is added only once. The pseudo-code explaining this pre-processing steps is shown in algorithm 1. This pre-processing step is mandatory regardless which chosen scoring method or tree construction strategy is executed.

This pre-processing step is crucial for the proposed approach since it relies on a predefined number of taxonomy levels. For example, if we have a document assigned to a fifth level category and the system operates only at the first two levels of the taxonomy, the approach will disregard this document. To prevent this, the ancestor completion step will make sure that the second level parent of the level-5 category is also included in the document's category list. In this situation the scoring step will generate a score for the second level parent of the document, and the tree construction phase will construct a final two level tree containing only relevant t-facets.

In addition to that, additional pre-processing steps might be required depending on which scoring method is used, these include:

1. Calculating document level statistics (Required if the LTR scoring method will be used):

Algorithm 1: Pre-processing Document's Categories

```
Input: document, taxonomy_tree
Result: Expanded categories list
categories_list = retrieve_categories(document);
;                                  // Returns assigned categories to this document
complete_category_list= {};
for each category ∈ categories_list do
    complete_category_list.append(category);
    ancestors_list= find_ancestors(category);
    ;    // Finds all ancestors of this node in the taxonomy tree and returns
     them, except for the root node
    for each ancestor ∈ ancestors_list do
        if ancestor not in categories_list then
            complete_category_list.append(ancestor);
        end
    end
end
Output: complete_category_list
```

(a) *Average rating*: Average rating values assigned to the document by users,

(b) *Ratings count*: number of ratings for a document,

(c) *Reviews count*: number of reviews for a document,

(d) and *Average sentiment*: Average sentiment score for the document's reviews.

2. Vector representation is produced and stored for all the t-facets labels in the system regardless of their level. The technique used for vector generation depends on the used scoring method. (Required if VSM in section 3.3.2 or LTR in section 3.3.3 scoring methods are used).

3. Vector representation is produced and stored for each document's textual content, this includes a separate vector representation for its textual description, and reviews text. (Required if VSM or LTR scoring method are used).

3.2.4 Formal Notations

A summary of the notations used in this thesis and their description are listed in table 3.1. It is an extended and modified version of the notations suggested by Schuth and Marx [67]. The new version re-defines the notations to fit type-based facet ranking context. In addition to that, it complements the original list with extra user related notations and vector based notation.

Term	Description
Q	Set of all input queries submitted to the system.
q	Current textual query being processed, $q \in Q$.
U	Set of the all users using the FSS.
u	Current single user being processed, $u \in U$.
D	Set of documents in the collection being searched in arbitrary order.
d	A single document in the dataset collection $d \in D$.
D_q	Ranked documents list, subset of D, retrieved and ordered by document search engine by relevance to the input query q.
D_U	Set of documents previously rated by at least one user of the system in U.
D_u	Set of documents previously rated by a single user u.
FT	Structured complete type-based facet taxonomy tree utilized by the FSS.
F	Flat list of type-based facets in the document collection D in arbitrary order, $F \in FT$.
F_q	List of type-based facets associated with documents which belong to D_q.
F_U	List of t-facets rated by at least one user in the system, i.e. t-facets which belong to D_U.
F_u	List of t-facets associated with documents rated previously by the user in D_u.
f_i	The current t-facet f to be ranked, where $i \in \{1, 2, 3, ..., \lvert F_q \rvert\}$, in filtering operation it keeps only documents associated with it.
k	The number of ranked documents displayed per page.
p	The number of facets displayed per facet page.
f_i^R	The current facet f_i relevance score, $f_i^R \in \{1, 0\}$. Where 1 denotes relevant facet and 0 denotes irrelevant facet.
$\overrightarrow{f}$	Vector representing the t-facet f.

Table 3.1: Description of notations used in the equations in this thesis.

3.3 Step 1: Personalized T-Facet Scoring Methods

As explained earlier, the t-facet ranking process is triggered after the document search engine retrieves the relevant search results. Then, the approach gathers the t-facets associated with the retrieved results. This initial set of t-facets is the input for the t-facet ranking process. Other t-facets in the types taxonomy are considered irrelevant and excluded from the ranking process.

The first step operates exclusively on the last level t-facets (leaf nodes). The scoring algorithm takes them as inputs and generates a relevance score for each individual t-facet. The score reflects both relevance to the user and the query. In the following sub-sections, three possible scoring methods are proposed to calculate this individual t-facet score. In chapter 5, the performances of the three methods are compared and discussed. The final outputs of this method are pairs of end level t-facets and their corresponding relevance score. The higher the score the more relevant it is to the user and the query, the lower the score the less relevant it is to them.

Let us consider the example of FSS configured to function on three level types taxonomy. The first step will only consider third level t-facets, it will calculate score for each one of them. The second step (explained in section 3.4) will use the returned scores and deal with the rest of the relevant t-facets. The remaining t-facets in this case include first and second level ancestors of the third level scored t-facets. The second step will group and sort the higher level t-facets in order to produce the final t-facet tree.

3.3.1 Using A Probabilistic Model

In this step, probabilistic models are used to estimate the t-facet relevance score given a query and a user profile. Probabilistic models provide theoretical models to estimate the probability that a document d is relevant to a query q. It is widely adopted in IR research. In this work, the proposed models are based on the well-known probabilistic models introduced by Sontag et al. for personalized web search [68]. The authors personalized search results using topic-based user profiles. The

user profiles are collected from users' historical interactions with the system. Their approach then re-weights the original document level search results according to topic relevancy to the user and query. In this work, we utilize the topic re-weighting factor to derive the t-facets score. The generated score reflects t-facet relevance to both the user and the input query. The usage of previous user interactions and topic-models make it an appropriate method to use for personalized t-facet ranking. Below we re-define those models in the context of our t-facet scoring task.

To generate the t-facet score, two models are proposed; ***Model 1*** assumes no background data available. It focuses on the individual user preferences without including any preferences from other users in the system. On the other hand, ***Model 2*** utilizes background data collected from system users in addition to the individual user preferences to estimate the relevance score.

Model 1 is calculated using the following formula:

$$score(f_i) = \sum_{f_u}^{F_u} P(f_u|q, \theta_u) \times P(cov(f_u, f_i)|f_u, f_i) \tag{3.1}$$

Where f_i is the current t-facet to be ranked, f_u is t-facets rated before by the user, θ_u is the user profile, and $P(cov(f_u, f_i)|f_u, f_i)$ is the probability that t-facet f_u is *covered* by the t-facet f_i. In other words, it reflects how much f_u is represented by f_i. We estimate it using two methods: 1) ***Exact*** match method: In which $P(cov(f_u, f_i)|f_u, f_i)$ equals to 1 if $f_u = f_i$, otherwise 0. 2) ***Cosine*** method: The authors of the probabilistic approach which this method is derived from, state that the probability can also be estimated using a function of distance between f_u and f_i. In our case, the distance function is derived from the cosine similarity between vectors generated for the input t-facet labels. The vectors are generated using a pre-trained generic BERT model. BERT is useful in embedding the category label while preserving its semantics. It was found useful in many IR tasks. Using it in this method gives us the flexibility to handle new and unseen t-facets.

We call the *Cosine* case a coverage measure, rather than a probability, because

this similarity method does not have the probability properties. Since we are using this formula for ranking purposes, relaxing this condition will not affect the ranking process.

Finally, the probability $P(f_u|q,\theta_u)$ models users' preferences towards the t-facet f_u. Details about how this probability is estimated are provided later in this section.

Now turning to ***Model 2***, the model uses background data collected from other users in the FSS to estimate the t-facet score:

$$score(f_i) = \frac{\sum_{f_u} P(f_u|q,\theta_u) \times P(cov(f_u,f_i)|f_u,f_i)}{\sum_f P_r(f|q) \times P(cov(f,f_i)|f,f_i)} \tag{3.2}$$

The numerator is the same as Model 1, in the denominator, the background distribution $P_r(f|q)$ (where r denotes a random or generic user) is calculated by averaging the relevance score for the top N search results belonging to this t-facet when the query q is submitted to a search engine. $P_r(f|q)$ can be obtained using the following equation:

$$P_r(f|q) = \frac{1}{N} \times \sum_{m=1}^{N} P(rel(d_m,q) = 1|q) \times P(f_d|d_m) \tag{3.3}$$

Note that in our case the probability that a document d belongs to a given t-facet f_d equals one, i.e. $P(f_d|d) = 1$, since venues' types are assigned by their owners, i.e. the type of the venue is not estimated, it is known. Details of the derivation of the two models can be found in [68].

Now we turn to modeling the users preferences, where we estimate $P(f_u|q,\theta_u)$. For this purpose, the method employs users historical ratings. It assumes that the user prefers t-facets of the venues they rated positively in the past. We use the generative

model suggested by Sontag et al. to estimate this value. In this model $P(f_u|q, \theta_u)$ is estimated using Bayesian rule:

$$P(f_u|q, \theta_u) = \frac{P(f_u|\theta_u) \times P(q|f_u)}{\sum_{f'} P(f'|\theta_u) \times P(q|f')} \tag{3.4}$$

Where $P(f_u|\theta_u)$ is estimated by dividing how many times the user rated documents belong to f_u positively, divided by the total number of documents rated by the user. The probability $P(q|f)$ is estimated by inverting $P_r(f|q)$ (see Eq.3.3 for how $P_r(f|q)$ is obtained):

$$P(q|f) = c\frac{P_r(f|q)}{P_r(f)} \tag{3.5}$$

Where c is a constant set to one since we are using this method for ranking and $P_r(f)$ is the probability that a random user rates this facet positively. $P_r(f)$ is obtained by counting how many times these t-facets documents were rated positively by all system users, divided by the total number of documents rated by them.

3.3.2 Using Vector Space Model and Rocchio Formula

This scoring method takes advantage of the t-facets label meaning. It extracts vector representation of the t-facet being ranked and uses the Rocchio algorithm to generate its relevance score. Rocchio relevance feedback algorithm is a classical approach in IR that implements relevance feedback in vector space models. It has been used in literature with varying goals, ranging from query expansion, document re-ranking to personalized document ranking in addition to other tasks [66, 69, 70, 71].

The relevance between the t-facet and the user profile is established through their vector representation. Each t-facet f_i is represented by a BERT embedding $\overrightarrow{f_i}$ (see Eq. 3.6). We employ the t-facets label to generate its corresponding BERT embedding using a pre-trained BERT model [72]. BERT provides a meaningful vector representation of text and has been proven effective in many IR tasks. For example, the t-facet 'Christmas Market' is given as input to a pre-trained BERT model, and

the output embedding is used as representation for this t-facet.

$$\overrightarrow{f_i} = BERT(f_i^{name}) \tag{3.6}$$

The vector representing the user profile is obtained using the Rocchio formula, which is a classical approach to relevance feedback in IR. Since in our case we only have the user feedback (i.e. ratings) at document level, we assume that the rating can also be transferred to the types associated with the document. Based on this assumption, we represent the user profile through a vector combining positive, negative and neutral preferences (i.e. rated document types). We define the positive (pos.) user vector $\overrightarrow{u_i}^{pos}$ as the average of t-facet vectors rated positively by the user u_i:

$$\overrightarrow{u_i}^{pos} = \frac{\sum_{j=1}^{|F_u^{pos}|} \overrightarrow{f_j} \times w_{f_i^{pos}}}{|F_u^{pos}|} \tag{3.7}$$

Where F_u^{pos} is the set of t-facets rated positively by the user. In our version of the user positive $\overrightarrow{u_i}^{pos}$, each vector is weighted by $w_{f_j^{pos}}$ (see eq 3.8). The weight averages the probability that this specific user will rate the t-facet as positive $Pr(f_j^{pos}|u_i)$ plus the probability that any user in the system will rate the t-facet as positive $Pr(f_j^{pos}|U)$. In the absence of individual personal preference, only the general probability can be used. The value $w_{f_j^{pos}}$ affects how each t-facet contributes to the final positive profile vector.

$$w_{f_j^{pos}} = \begin{cases} \gamma \times Pr(f_j^{pos}|U) + (1-\gamma) \times Pr(f_j^{pos}|u) & \text{, if } Pr(f_j^{pos}|u) > 0 \\ Pr(f_j^{pos}|U) & \text{, otherwise} \end{cases} \tag{3.8}$$

The parameter $\gamma \in [0, 1]$, is used to balance between individual user feedback versus general population feedback.

In a similar manner, the negative (neg.) user vector is computed as follows:

$$\overrightarrow{u_i}^{neg} = \frac{\sum_{j=1}^{|F_u^{neg}|} \overrightarrow{f_j} \times w_{f_i^{neg}}}{|F_u^{neg}|} \tag{3.9}$$

In case neutral document ratings exist in the FSS, the user neutral (neu.) vector calculation follows the same equation:

$$\overrightarrow{u_i}^{neu} = \frac{\sum_{j=1}^{|F_u^{neu}|} \overrightarrow{f_j} \times w_{f_i^{neu}}}{|F_u^{neu}|} \tag{3.10}$$

In order to model the final vector representation of the users interest, we experimented with two versions of the Rocchio formula. The first version called ***VSM-Rocchio***. It employs the traditional version computed according to equation 3.11.

$$\overrightarrow{u_i} = \alpha \times \overrightarrow{u_i}^{neu} + \beta \times \overrightarrow{u_i}^{pos} - \lambda \times \overrightarrow{u_i}^{neg} \tag{3.11}$$

The contribution of each user vector is regulated through the weights $\alpha, \beta, \lambda \in [0, 1]$. During experimentation, varying weights are examined to find the best combination to improve the final t-facet score results.

Basile et al. [69] proposed an improved version of the Rocchio formula, in which instead of subtracting the negative vector, they added the users negative vector's orthogonal complement ($\overrightarrow{u_i}^{neg^{\perp}}$). In this work, we include this version in our experimentation to see how it can improve the results. We call this version ***VSM-Ortho***, Equation 3.12 illustrates the computation. The orthogonal complement is obtained

using Gram–Schmidt process. The weights are the same as equation 3.11.

$$\overrightarrow{u_i} = \alpha \times \overrightarrow{u_i}^{neu} + \beta \times \overrightarrow{u_i}^{pos} + \lambda \times \overrightarrow{u_i}^{neg^{\perp}} \tag{3.12}$$

Finally, the query t-facets are ranked according to the cosine similarity score between the user profile vector $\overrightarrow{u_i}$ and the vector representing the t-facet $\overrightarrow{f_j}$ according to the following equation:

$$score(f_j, u_i) = cosine\_similarity(\overrightarrow{u_i}, \overrightarrow{f_j}) \tag{3.13}$$

Note the following,

- This is pure personalized t-facet ranking, meaning that it does not take the current input query into account in the scoring directly. However query relevance is achieved in an indirect way, because after the document ranking happens, only t-facets associated with the relevant documents are considered in this step. This ensures that each t-facet has at least one document marked as relevant by document search engine.
- The approach has some advantages, the distances for all t-facets can be pre-computed and stored with the user profile making this a less complex and faster approach. The user profile vector also can be pre-computed and updated periodically as the user rates new documents.
- Through its usage of BERT, this scoring method leverages the categorical characteristics of t-facets. This is especially useful in handling unseen t-facets automatically (i.e. no need for special handling procedure for this case). In case of unseen or new t-facets the system will generate the embedding and calculate similarity normally.

3.3.3 Using Deep Learning To Rank Model

This scoring method is based on the intuition that there are multiple 'signals' surrounding the user's interaction with the system that capture the relevance of a given t-facet for a user, given their history and context. We then define the problem in terms of learning to rank t-facets based on a number of features, each capturing a different interaction aspect. A Deep Neural Network (DNN) model is trained over these features, to capture the intricacy of the relationship between these features and the users profile and interests.

In order to generate the t-facet relevance score, this method models the problem of ranking t-facets as a learning to rank problem. For each user and query, it collects a set of features aiming to capture the deemed relevance of a given t-facet. Three groups of features are computed for each t-facet, query and user tuple. ***Group-CF*** contains collaborative filtering features, ***Group-P*** contains personalization features and ***Group-Q*** includes features reflecting the query relevance. Details of how each group of features are calculated are provided in the next subsections.

The three groups (Group-CF, Group-Q and Group-P) of features are then fed into a deep neural network trained to predict a t-facet ranking score. DNN models have shown to be effective to combine many useful features into a ranking score. The higher the score, the higher the relevance of the t-facet to both the user and the query. In the next section the input features are described in detail, followed by the DNN architecture built to train the final prediction model.

Personalization Features (*Group-P*)

This group of features incorporates individual user preferences into the ranking process. All features in this group are calculated on the facet level. Previous user ratings are used to build a preference profile for each user. When the user rates a visited document positively, it is assumed that the user also likes its categories. The same also applies for documents rated as neutral or negative. Based on this assumption, we compute the following features:

- ***uf_positive_prob***: The user facet positive probability is the probability that the user rates the t-facet as positive in general, calculated using the following formula:

$$uf_i\_positive\_prob = \frac{freq^+(f_i, F_u)}{|F_u|} \tag{3.14}$$

Where $freq^+(f_i, F_u)$ is the frequency of the t-facet f_i in the set of t-facets rated positively by the user F_u. Divided by the count of all t-facets rated by the user.

- ***uf_positivity_rate***: the user facet positivity rate. It is the probability that the user rates this t-facet as positive when she/he sees a document associated with it. It is different from the previous feature as it considers only the ratings given by the user to this specific t-facet, obtained using this equation:

$$uf_i\_positive\_rate = \frac{C^+(u, f)}{C(u, f)} \tag{3.15}$$

The equation divides the count of how many times the user rated this t-facet as positive $C^+(u, f)$ by the number of times the user rated in general this specific t-facet $C(u, f)$ (this includes positive, negative and neutral ratings). It is important to include this feature along with the previous one as it is very likely to have the same t-facet associated with two separate documents, one of them rated positively and the other negatively.

The last two features together reflect whether the user has a strong attitude towards the t-facet on its own, or whether it depends on the individual document content.

- ***uf_negative_prob***: The user facet negative probability. It is the probability that the user rates the t-facet as negative in general, calculated using the same equation 3.14, but counts the number of times when the t-facets are rated as

negative. $freq^{-}(f_i, F_u)$ in this case is frequency of the t-facet f_i in the t-facets rated negative by the user. This value is then divided by the number of all t-facets rated by the user.

- ***uf_negativity_rate***: the user facet negativity rate. It is the probability that the user rates this t-facet as negative when they see it. It is different from the previous feature as it considers only the ratings given by the user to this specific t-facet, equation 3.15 can be used to compute this feature, but for negative t-facets. It divides the count of how many times the user rated this t-facet as negative by the number of times the user rated this specific t-facet.

- Similarly, the features for the neutral t-facets, ***uf_neutral_prob*** and ***uf_neutrality_rate***, are also obtained.

- In addition to these, other features characterizing the user profile are also added to this group:

  - ***user_gender***, ***user_age_group***. User age is mapped into fixed intervals (for example 10 to 15 years). Both gender and age groups reflect the group interest in a specific t-facet. For example kids will not be interested in bars.

  - Additional features like ***user_fans_count***, ***user_avg_ratings***, ***user_reviews_count*** are also added to enrich the user profile (if this information is available in the data collection).

The features are calculated at the t-facet level for each individual user and t-facet pair. They can be pre-calculated offline for all the t-facet taxonomy and stored in the user profile in advance. This profile will be updated as the user rates new POIs. When the user issues a new query, the search engine retrieves the relevant places. This method will then get the pre-calculated features from the user profile and combine them with the other group features as input to the DNN model.

Collaborative Filtering Features (*Group-CF*)

This group of features reflects the user feedback of a collective of users about the given t-facet. Similarly to the user-based features, we compute the following list of collaborative-based features:

- ***cf_positive_prob***: The collaborative t-facet positive probability. It is the probability that users rate the t-facet as positive in general, calculated using the following formula:

$$cf_i\_positive\_prob = \frac{freq^+(f_i, F_U)}{|F_U|} \tag{3.16}$$

  Where $freq^+(f_i, F_U)$ is frequency of the t-facet f_i in the t-facets rated positive by all the users. Divided by the count of all t-facets in the system $|F_U|$, where U is the set of all system users.

- ***cf_positivity_rate***: the collaborative facet positivity rate. It is the probability that the users rate this t-facet as positive when they see it.

$$cf_i\_positive\_rate = \frac{C^+(f_i, F_U)}{C(f_i, F_U)} \tag{3.17}$$

  The equation divides the number of times users rated this t-facet as positive $C^+(f_i, F_U)$, with how many times the users rate this specific t-facet $C(f_i, F_U)$.

- ***cf_negative_prob***: The collaborative facet negative probability. It is the probability that users rate the t-facet as negative in general, calculated using the same equation 3.16, but counts the number of times when t-facets are rated as negative. $freq^-(f_i, F_u)$ in this case is frequency of the t-facet f_i in the t-facets rated negative by all users. This value is then divided by the number of all t-facets in the system.

- ***cf_negativity_rate***: the global t-facet negativity rate. It is the probability that users rate this t-facet as negative when they see it. Equation 3.17 can be used to compute this feature, but for negative t-facets. It divides the count of how many times the user rated this t-facet as negative, with how many times the user rates this specific t-facet.

- In the same way, two features for neutral t-facets ***cf_neutral_prob*** and ***cf_neutrality_rate*** can also be obtained.

The features are also calculated on the t-facet level, and can be pre-computed offline and stored in a global user profile and updated periodically. Depending on the FSS domain and context, system admins can decide whether to generate one universal reference profile for all the users, or to segment the user base according to their demographics (like age group, geographical location...etc.). In either case, this group of features reflect the general searchers attitude towards the specific t-facet.

In addition, additional collaborative filtering features commonly used in literature are extracted for each relevant document and averaged on the t-facet level, as listed below:

- ***avg_rating***: Average ratings of relevant documents associated to this t-facet.

- ***avg_rating_count***: Average count of ratings the relevant documents associated with this t-facet received.

- ***avg_reviews_count***: Average of the number of ratings the relevant documents received.

- ***avg_reviews_sent***: Average reviews polarity. In order to calculate this feature, sentiment analysis is performed for the most recent reviews of the document. Then, the sentiments are averaged into one overall polarity score for the document. This overall sentiment per document is then averaged on the t-facet level.

- ***price_group***: This feature rates the price group to which this document

belong. For example, in POI suggestion task, some users prefer cheaper places or more expensive ones which should be considered during the ranking process.

- ***avg_cat_count***: The category count (i.e. the different number of category types associated with this document). Only end level categories are considered for this count. For example, a food place providing more than one food type might be ranked higher.

- ***avg_cat_depth***: It reflects the depth of the category in the hierarchy type tree. Category depth considers the number of all categories regardless their level. This includes the parent categories as well. If the document belongs to more than one category, the sum of the depth of all categories is taken. Common ancestors are added only once. For example, a club specialized in specific sport might rank higher.

These features are calculated at document level and are aggregated by averaging their values on the t-facet level as explained earlier. Only relevant documents which belong to this t-facet are considered.

Query Relevance Features (*Group-Q*)

The features in this group reflect the relevance of the t-facet to the input query and to the set of relevant results returned by the search engine in response to it. The features generated at the t-facet level are:

- $\boldsymbol{max\_sim(f_i, q)}$: Maximum cosine similarity between t-facet name and each keyword in the query. This feature seeks to capture the direct mention of the category in the query. As the similarity approximates 1, this means the category is explicitly mentioned in the query. When this happens, the t-facet relevance should be boosted.

- $\boldsymbol{avg\_sim(f_i, q)}$: Average semantic similarity between the t-facet and each keyword in the query. If multiple keywords are similar to the t-facet this will result in higher average and ultimately higher t-facet importance. The semantic

similarities are computed using the cosine between the BERT vectors representing the two input texts.

- Another set of features are related to the t-facet information gain. These measure how much information is gained when the user selects this t-facet. This feature set employs the document-level search engine score in several ways to compute these scores. It assumes that the relevance travels from the documents to their t-facets:

  - ***info_gain***: the average total score of all the relevant documents associated with this t-facet.

$$info\_gain = \frac{\sum^{D_q} score(d, q)}{|D_q|} \tag{3.18}$$

  - ***mutual_info_gain***: similar to the previous one, but calculated only on documents which are unique to the t-facet. Unique means documents which are only associated with this t-facet. A greedy approach is followed to calculate this feature. It ranks the t-facets first by count, then considers documents unique to the first t-facet and they are added to its gain. Then it moves to the next t-facet, excludes already seen documents, calculates average $mutual\_info\_gain$ for the rest and so on. This feature highlights the unique importance of documents which belong only to this specific t-facet. The procedure is illustrated in algorithm 2.

  - ***info_gain*@1**: Information gain at 1 is the score of the top ranked document seen by the user at first position, when they filter the results using this t-facet. This feature highlights t-facets associated with a document with a high relevance score.

  - ***info_gain*@*k***: Information gain at k is the average for the k top documents seen by the user at the top of the result page, after they filter the results using this t-facet. k is the size of the search result page. The feature favors t-facets with many highly ranked documents in the first page;

which increases the chances that the user will find the target document and fulfill their search needs.

- ***popularity*(f_i, F_q)** is the t-facet popularity among the facets associated with the query results. It is calculated by counting the number of relevant documents that belong to this facet type divided by the total number of documents in the retrieved result set, see equation 3.19. This feature is inspired by the most common used facet ranking method in literature [3].

$$popularity(f_i, F_q) = \frac{freq(f_i, F_q)}{|F_q|} \tag{3.19}$$

- ***mutual_popularity*(f_i, F_q)** is the t-facet unique popularity among the t-facets associated with query results. It is calculated by counting the number of unique relevant documents that belong to this facet type divided by the total number of documents in the retrieved result set. It is calculated using a greedy approach like ***mutual_info_gain*** feature, but instead of adding the document score, it counts the number of unique documents belonging to this t-facet.

Algorithm 2: Greedy approach to calculate mutual info_gain feature.

```
Input: ranked_document_set
Result: A set of t-facets and their mutual information gain values
t_facets_list = retrieve_categories(relevant_document_set);
;           // Returns assigned end leaf categories to relevant document set
info_gain_t_facets={};
; // Set contains tuples of t-facets and their final mutual info_gain values
while relevant_document_set is not empty do
    <best_t-facet,best_info_gain>=get_max_gain_t-facet(
     t_facets_list,relevant_document_set);
    ;    // Retrieve the t-facet with highest average score for the remaining
     document set.
    covered_documents=
     filter_documents(best_t-facet,relevant_document_set);
    relevant_documents_set.remove( covered_documents);
    t_facets_list.remove(best_t-facet);
    info_gain_t_facets.append( <best_t-facet,best_info_gain>);
end
for t-facet ∈ t_facets_list do
    info_gain_t_facets.append( t-facet, 0 );
    ;   // Adding the remaining t-facets with info_gain equals to zero (means
     no new gain by filtering using this t-facet)
end
Output: info_gain_t_facets
```

DNN Model Architecture

The employed DNN model is used as pointwise Learning To Rank (LTR) algorithm, we call this scoring method (***DNN-LTR***). In order to build the Deep Neural Network (DNN), each group of features is fed into a separate sub-network first. The sub-network role is to reduce the features to a fixed number of nodes n per network. Layers within the same sub-network are fully connected. The sub-networks are disconnected from each other.

Regardless of the different number of features that the group contains, the group's sub-network will encode the input in n nodes. These sub-network outputs are then

concatenated in a single layer and used as inputs in the final prediction network. During the training process the final network will utilize the inputs from each sub-network node, thus its corresponding feature group, to predict the final t-facet relevance score.

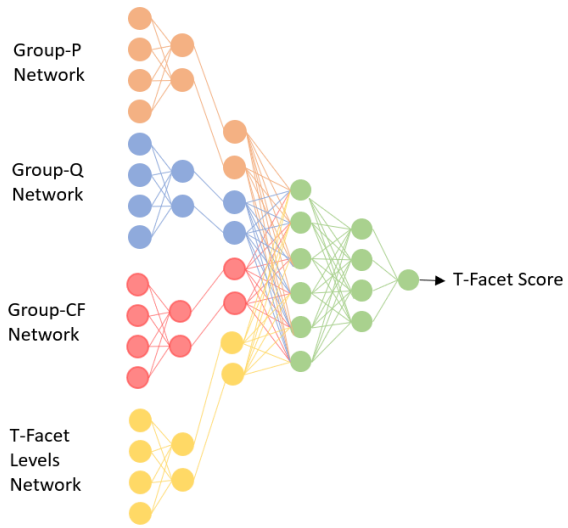


Figure 3.3: Illustration for the DNN architecture.

Figure 3.3 demonstrates the suggested architecture. The groups P, Q and CF features are passed first to the orange, blue, and red sub-networks. In addition the t-facet and its ancestors names are encoded using one-hot encoding used as inputs to the yellow sub-network. The yellow sub-network encodes t-facets into dense vector layers, which acts as t-facet identifiers during the training process.

The overall model takes a pyramidal shape, at each sub-network, each layer contains less number of nodes than the preceding one till it reaches the needed n nodes. The overall DNN is optimized using the Adam algorithm [73]. Regardless of the number of features in the input layer, the four sub-networks end with a layer containing exactly n nodes ($n = 2$ in the figure).

This architecture ensures an equal contribution from each feature group to the final scoring network, represented in green in the figure 3.3. It also avoids cases where the groups containing a large number of features dominate the training process. On the

other end, the final network (green) determines the appropriate way to combine the group features into a single t-facet relevance score.

For example, each t-facet vector contains more than a thousand hot-encoded input features, while other feature groups, like Group-Q, contain less than 10 features. In this case, important signals from the Group-Q features can be out-weighed by the t-facets features. Having the same number of nodes from each network ensures a balance between weights representing the t-facet relevance from each group perspective.

Deep Learning To Rank Method Summary

Table 3.2 summarizes the features used in this approach, their group and whether they are calculated on t-facet or document level. Finally, a vector is computed with features from all the three groups. The feature vector includes also the t-facet and its ancestors names added as categorical features.

The features are extracted only for t-facets associated with relevant documents returned by the search engine after the user submits the query. The DNN model is trained on historical data from the FSS. The trained model predicts the final t-facet relevance value.

Feature calculations are straightforward, they can either be pre-calculated and stored or use pre-stored embedding. No complex heavy calculations are needed, but yet the features reflect the relevance effectively. This will help make the prediction lighter and faster at run-time, without compromising the ranking process efficiency.

Group	Feature	Short Description
Group-Q	$\boldsymbol{max\_sim(f_i, q)}$	Maximum cosine similarity between query keywords vectors and t-facet name vector.
	$\boldsymbol{avg\_sim(f_i, q)}$	Average cosine similarity between query keywords and the t-facet vector.
	$\boldsymbol{info\_gain}$	Average score of relevant documents belonging to this t-facet.
	$\boldsymbol{mutual\_info\_gain}$	Average score of relevant unique documents relevant to this facet, see algorithm 2.
	$\boldsymbol{info\_gain@1}$	The top relevant document belonging to this t-facet score.
	$\boldsymbol{info\_gain@k}$	Average of top k relevant documents belonging to this t-facet score, where k is the document results page length.
	$\boldsymbol{popularity(f_i, F_q)}$	Number of documents associated with the t-facet divided by total count of relevant documents.
	$\boldsymbol{mutual\_popularity(f_i, F_q)}$	Number of unique documents associated with the t-facet divided by the total count of total documents.

Table 3.1 Extracted DNN-LTR Features Summary. – continued on next page

Group	Feature	Short Description
Group-P	***uf_positive_prob***	Probability that current user will rate t-facet as positive given all other t-facets ratings.
	uf_positivity_rate	Probability that current user will rate t-facet as positive given how many times the user rated this t-facet.
	uf_negative_prob	Probability that current user will rate t-facet as negative given all other t-facets ratings.
	uf_negativity_rate	Probability that current user will rate t-facet as negative given how many times the user rated this t-facet.
	uf_neutral_prob	Probability that current user will rate t-facet as neutral given all other t-facets ratings.
	uf_neutrality_rate	Probability that current user will rate t-facet as neutral given how many times the user rated this t-facet.
	user_gender	The current user gender.
	user_age_group	The current user age group.
	user_fans_count	Number of system users following the current user.

Table 3.1 Extracted DNN-LTR Features Summary. – continued on next page

Group	Feature	Short Description
	user_reviews_count	Number of reviews given by user.
	user_avg_ratings	Average rating value given by user.
Group-CF	***cf_positive_prob***	Probability that users globally will rate t-facet as positive given all other t-facets ratings.
	cf_positivity_rate	Probability that users globally will rate t-facet as positive given how many times the user rated this t-facet.
	cf_negative_prob	Probability all users globally will rate t-facet as negative given all other t-facets ratings.
	cf_negativity_rate	Probability that users globally will rate t-facet as negative given how many times the user rated this t-facet.
	cf_neutral_prob	Probability that users globally will rate t-facet as neutral given how many times the user rated this t-facet.
	cf_neutrality_rate	Probability that users globally will rate t-facet as neutral given how many times the user rated this t-facet.

Table 3.1 Extracted DNN-LTR Features Summary. – continued on next page

Group	Feature	Short Description
	avg_doc_rating	Average overall rating for relevant documents associated with this t-facet.
	avg_rating_count	Average ratings count for relevant documents associated with this t-facet.
	avg_reviews_count	Average reviews count for relevant documents associated with this t-facet.
	avg_reviews_sent	Average reviews sentiment for relevant documents associated with this t-facet.
	avg_reviews_sent_weighted	Average reviews sentiment for relevant documents associated with this t-facet weighted by the up votes given by users to the review.
	avg_cat_count	Average end-level categories for relevant documents belonging to this t-facet.
	avg_cat_depth	Average total categories for relevant documents belonging to this t-facet including the ancestors.
	$\boldsymbol{popularity(f_i, F_q)}$	Number of relevant documents associated with t-facet.

Table 3.2: Extracted DNN-LTR Features Summary.

In the experimental results chapter (section 5.2.3), the effect of each group of features on the final t-facet rank is investigated. This is achieved by evaluating the DNN-LTR scoring method using all features combined, then re-evaluating using each group of features in isolation, then the evaluation will be repeated for different combinations of the feature groups.

3.4 Step 2: T-Facet Tree Construction

As explained earlier, the proposed overall approach uses scoring methods to assign a score to each last level t-facet individually. The tree construction algorithm re-orders the original taxonomy tree by using the generated scores. It follows a bottom-up approach, where the t-facets at the lower level in the taxonomy are sorted first, then it proceeds in sorting all the ancestors of those facets.

In this section we introduce several alternative strategies to achieve this ranking. These strategies control two things: 1) How to rank the current level of facets given the ranking of the ones in the previous level, and 2) It also decides the t-facet sub-tree that will appear to the user at the first facets page. Remaining facets will be available to the user by clicking the 'More' link. Regardless the chosen strategy, the system starts to rank the lowest level of the facets first. Then it moves up and ranks the higher level and this is repeated till it reaches the top level. At each level, the ranks of the previous level are employed to induce the ranks of the current level.

Recall, from the data structure section, that the system admin configures how many levels from the taxonomy tree will be displayed to the user. They also configure the maximum t-facet page length, which is how many t-facets will appear to the user at the landing page, and any consecutive pages if required. It is up to the strategy to decide how many t-facets from each level will be displayed. It can divide the page length equally between levels, for example if the total t-facet page size is 9, the strategy might enforce the t-facet tree to contain three level-1 facets, and for each one it should contain three level-2 facets. Or it might decide to show the the top nine level-3 t-facets and their parents, and so on.

```
Input t-facet list from
scoring phase:

Cat A, subCat A.5, score=10
Cat D, subCat D.8, score=9
Cat B, subCat B.4, score=8
Cat D, subCat D.1, score=7
Cat F, subCat F.8, score=6
Cat B, subCat B.1, score=5
Cat D, subCat D.2, score=4
Cat B, subCat B.2, score=4
Cat A, subCat A.3, score=3
Cat A, subCat A.4, score=2
Cat D, subCat D.3, score=2
Cat G, subCat G.20,score=2
Cat B, subCat B.7, score=1
Cat B, subCat B.3, score=1
Cat G, subCat G.2, score=1
```

Figure 3.4: Example for input t-facet list from scoring phase, Number of levels=2.

In order to demonstrate the suggested strategies, an example input scored t-facet list in figure 3.4 will be used in all strategies. It operates on 2 level categories taxonomy, and contains 5 relevant level-1 t-facets and 15 relevant level-2 t-facet . For simplicity we call level-1 facet categories denoted by the abbreviation 'Cat' in the figure, where level-2 types are called sub-categories denoted by 'subCat' in the same figure. In this example the level-2 t-facets' score ranges between 1 and 10, with 10 being the most relevance t-facet to the query and the user, and 1 the least relevant t-score facet for both user and query.

It is important to stress that this research is concerned with t-facet tree construction from metrics improvement perspective. It is acknowledged that there are other HCI factors that are involved in the decision of which strategy is to be followed in this phase. In this work, the conducted experiments will study how the strategy decision will affect the effort the user needs to put in, in order to fulfill her/his search needs. This effort is measured using several evaluation metrics introduced and discussed in

chapter 4.

3.4.1 Strategy 1: Plain List - No Grouping Baseline

The strategy takes the list of facets ordered by the score and displays it as is to the user. No grouping based on the parent levels happens. Example of how this strategy works and its output is shown in figure 3.5. As a matter of fact, if the tree construction step is removed, this output would have been presented to the user. To highlight why this step is needed and how it adds value from metrics improvement perspective,this strategy will serve as a baseline to compare the other strategies against. This comparison will highlight the need for appropriate tree construction method and it will measure the change achieved by other strategies.

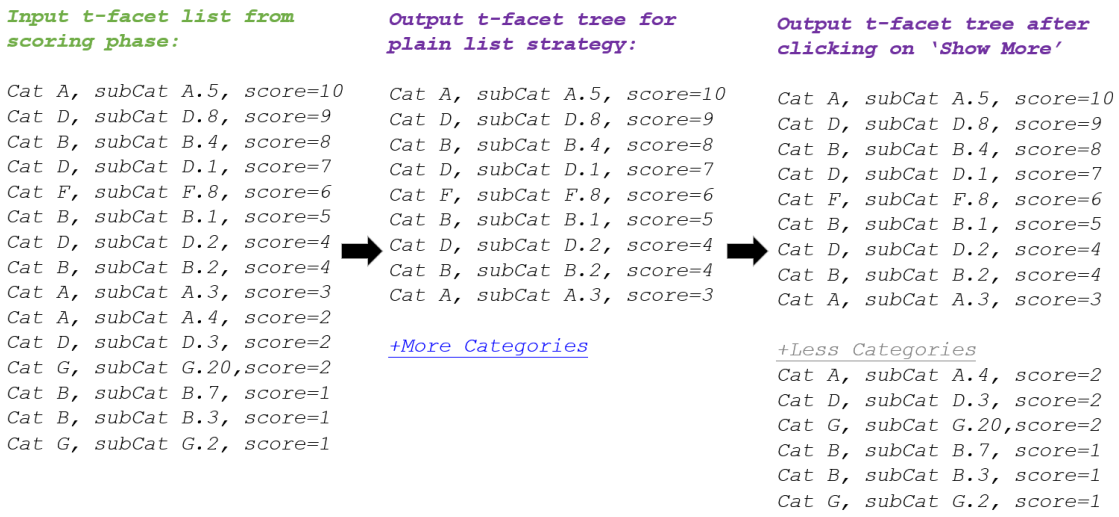


Figure 3.5: Constructing final t-facet tree following plain list strategy.

In the example shown in 3.5, the t-facet page size is limited to nine t-facets [2]. This strategy shows the top nine pairs of level-1 and level-2 t-facets and orders them according to the level-2 t-facets score without grouping children belonging to the same parent. The rest of the relevant t-facets are available via the 'More Categories' link,

[2]How many t-facets are to be displayed to the user per page is a predefined value by system admins and serves as an input to this approach, how it is decided is out of the scope of this research.

each consecutive t-facets page will contain the same number of t-facets like the first page till the t-facets end.

On the advantages side, theoretically this method should produce best metrics results, since the sub-categories with high relevance are ranked first. However, it is not practical to be shown to the users as you can see from the example that a couple level-2 sub-categories exist in the top 9 list for the same parent level-1 category A, one at the top of the list and the other at the bottom of the list. It is not logical to show them separately to the user. The same case happened with categories D and B. This approach does not respect the tree nature of the types taxonomy, which might confuse users.

3.4.2 Strategy 2: Plain List With Level Grouping

This strategy is very similar to the first one, it takes the input scored t-facet list, selects the top t-facets for the first page, then if two or more level-2 t-facets belong to the same level-1 parent, it groups them together. After that, it sorts level-1 nodes according to their top level-2 t-facets score, and for each parent it sorts its children by their score.

In the case where level-1 has additional relevant children t-facets but are not displayed in the first page, they will be available to the user after selecting '+More Cat X' link. Where other relevant level-1 t-facets will be available after clicking '+ More Categories' link. Each consecutive t-facets page will be sorted in the same way.

Figure 3.6 shows an example for this process. This strategy respects the original taxonomy hierarchy, and shows the user a more organized and readable t-facet tree. On the cons side, in some cases the strategy will result in a tree which displays lower score level-2 t-facets first. For example in figure 3.6, the strategy's final tree presents the sub-category A.3 with scores equal to 3 at second position, where all the rest of the displayed level-2 facets have a higher score. The same with sub-category D.1 with score 7, and sub-category D.2 with score four, both came before sub-category B.4 which has a higher score equal to 8. In order to show a more organized user-friendly

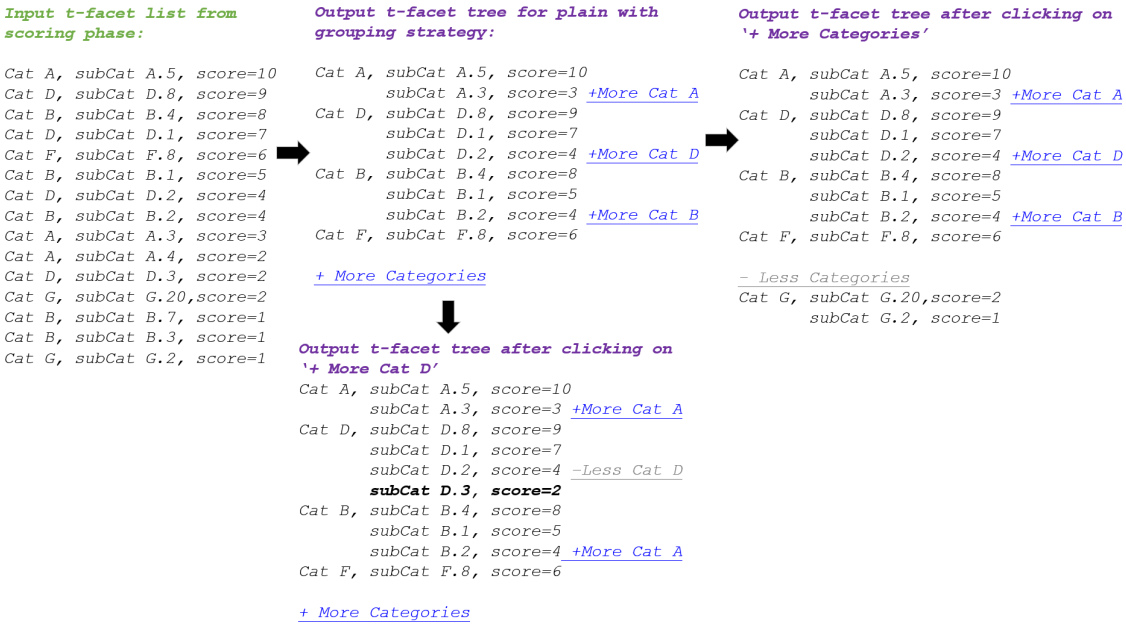


Figure 3.6: Constructing final t-facet tree following plain list with level grouping strategy, t-facet page size=9.

tree the strategy has to sacrifice the relevancy score ordering. This certainly will have an impact on the metrics measuring user effort, in order to reach the intended search target.

3.4.3 Strategy 3: Fixed Level Grouping

To build a final t-facet tree with v levels, the ***fixed level*** strategy follows a bottom-up approach. The strategy respects the original taxonomy hierarchy and uses a predefined fixed page size for each t-facet level. It starts by grouping t-facets at level-v by their parent. Then, it sorts the (parent) nodes at level-$(v-1)$ by aggregating the scores of their top k children, the children are ordered by their relevance score generated in step 1, and so on up to level-1. Several aggregation functions can be used, in our experiments we use average (Avg) and maximum (Max) functions.

Figure 3.7 shows an example for this process, categories (Cat.) correspond to level-1 t-facets and sub-categories (subCat) correspond to level-2 t-facets. In the case

where a level-1 facet has additional relevant t-facet children that are not displayed in the first page, they will be available to the user through the '+ More Cat ...'. Each following t-facet page will be sorted in the same way. The final output provides the user a more organized and readable t-facet tree.

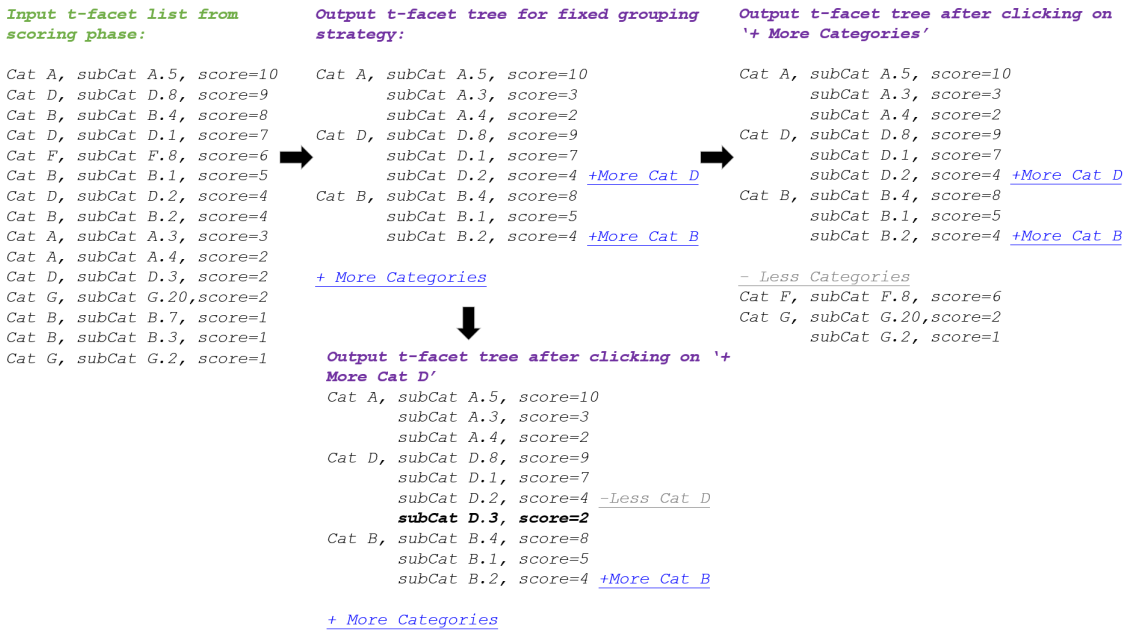


Figure 3.7: Constructing final t-facet tree following fixed level- Max grouping strategy, level-1 t-facets page size=3 and maximum of 3 level-2 t-facets per parent.

This approach diversifies the output tree, and prevents any level-1 category from dominating the list, giving a chance for other categories to appear in the first page. But like the previous strategy, due to enforced level grouping, some less relevant sub-categories might be promoted, however, this will be controlled by the fixed children count. In the example shown in figure 3.7 we can see that category F disappeared completely from the first page, because of the fixed number role sub-categories like A.4 were ordered higher in the tree. If the user was interested in category F, she/he will have to put in more effort and click on more category links to reach category F.

3.5 Approach Summary

This chapter proposed a two step approach to solve the t-facet ranking problem. The first step is a scoring step, which takes a list of relevant t-facets and assigns a numerical score to each leaf-node t-facet in it. Three alternative scoring methods were presented and explained in detail. The proposed methods achieve personalization and query relevancy by different means.

The output of this step is then processed by the tree construction step to build the final t-facet tree to the user. Several alternative strategies were suggested for this step. Currently, strategies adopted by the tree construction step are user and query independent.

More advanced or sophisticated variations of the tree construction strategies can also be used in the second step. Because this is a new research area we are focusing on separating this step from the scoring one, highlighting its importance, and how it contributes to the overall t-facet ranking approach.

Chapter 4

Experimental Design

A large and growing body of literature has investigated faceted ranking methods, however none introduced a well established bench-marked dataset with relevance judgments for facets, not to mention the special case of personalized type-based facets. In order to assess the effectiveness of the proposed methods, this chapter introduces a dataset customization framework which transforms existing IR collections for this purpose. The framework is used for generating datasets which are suitable for personalized t-facet task evaluation. In section 4.1, the framework describes the eligibility criteria for the dataset and domain, as well as the customization process applied. This is followed by a demonstration of how the framework can be applied to two different datasets which meet the eligibility criteria.

The developed framework, in combination with existing well-known simulated user interaction models and a number of IR metrics, are adopted to evaluate such a system. As the chapter dives further through the experimental design in detail, it lists the exact parameters chosen for the proposed approach and how they are selected in section 4.2.

Because of the novelty of the research task, there are no t-facet ranking methods which can be directly compared to this work's approach. Instead, baselines are

collected from key facet ranking methods in literature. The improvement of the proposed approach will be measured by comparing its results against those baselines, they are listed in section 4.3.

4.1 Dataset Customization Framework

In general, existing literature seems to follow two different paths to obtain evaluation collections with ground-truths in faceted search. The first is to utilize existing ad-hoc IR datasets with relevant judgments provided on the document level. In this case, it is assumed that relevance transfers from the documents to the facets to which they belong. This is the path followed by the INEX 2011 Data Centric Track [1]. The task consisted of two sub-tasks: an ad-hoc search task and a faceted search one. In the faceted search track, the evaluation metrics measured the effort needed to reach the first relevant result. The evaluation was based on the user simulation interaction model originally proposed by Koren et al. [14]. We follow this path in transforming the TREC-CS 2016 dataset in section 4.1.3. Our framework customizes the dataset to fit the type-based facet ranking task, existing personalized relevance judgments were useful to evaluate the facet ranking approach based on the same INEX 2011 Data Centric track assumptions.

The second path is to transform existing real-life datasets to fit facet ranking evaluation. In this case, the data collection does not provide explicit relevance judgments on the document level, instead they are being derived from historical users ratings. This path was followed by Koren et al. [14] on the MovieLens evaluation. In order to generate query requests, they used the most recent users ratings as search targets. The simulation approach was used to measure the user effort to reach those targets. As mentioned earlier in chapter 2, The MovieLens dataset is not suitable for our task as movie genres (types) are limited and do not have a multilevel hierarchical taxonomy. However, the methodology can be followed on other datasets.

Following the steps in this second path, our framework customizes real-life collections into a TREC-like format and then applies the INEX evaluation method. The

customization framework formalizes the dataset generation process and extends it to suit the case of type-based facets. As an example of the second path, we adopt the Yelp dataset described in section 4.1.4. In both paths, the same criteria needs to exist in the dataset in order to be a good fit for this research task, these are discussed in next section (Sec. 4.1.1).

4.1.1 Eligibility Criteria

It is important to start by identifying the criteria for selecting a suitable domain and dataset for this research topic. There are several domains to be explored, which has a faceted search nature such as product shopping, art collection search, museum visit planning, venue suggestion, and social event search. The applicability of the proposed framework is subjected to a number of criteria that pertain the domain, search task and type of data, listed as follows:

1. The underlying data collection is structured. It contains objects and each object has properties and types, which can be used as type-based facets.

2. The searchable objects belong to a rich taxonomy of categories, from which stems the need for ranking. This is a crucial requirement, as the categories act as type-based facets.

3. The data contains users feedback, ratings and reviews, which are useful for personalization. The more available user data the better the dataset collection. This is because the quantity of users related information impacts the quality of personalization models.

4. The data is accessible and available online, as some datasets need further data collection or have no pre-defined taxonomy of types, which makes them unsuitable for this task.

5. The dataset's domain should be suitable for faceted search, e.g. product shopping, digital libraries, venue suggestion, social event search, etc. This can be confirmed by existing FSS literature.

6. There is room for personlization in search result ranking (document level ranking), therefore personalization can also be applied on the facet ranking level. Existing literature related to the deemed search task and domain can be consulted to assess whether personalization plays a role in search result ranking or not.

The proposed framework is concerned with search tasks which aim at minimizing user effort in precision-oriented FSS. The assumption is that the search task is fulfilled as soon as the user finds their intended search target. The tourism domain, specifically the point of interest (POI) suggestion task is chosen for the search task. In addition to the availability of several online datasets, POI suggestion is a well-known personalization task for which it has already been proven that POI categories play an important role [64, 65].

Two datasets under this domain were selected and customized for experimentation, the first is ad-hoc IR TREC-CS dataset and the second is Yelp dataset. How this framework is applied in each case is shown later in this chapter. The datasets description, their statistics, how they meet the criteria, and how they were customized to fit the t-facet ranked process are covered in section 4.1.3 for the TREC-CS dataset, and section 4.1.4 for the Yelp dataset.

4.1.2 Generating Evaluation Requests

Typically, existing bench-marked IR datasets already contain requests and their relevance judgments at document level. In addition, datasets like the TREC-CS, provide the current search context and the users historical ratings. However, datasets adapted from real-life require an additional step to create requests that imitate this type of information.

To achieve this, user information including user historical picks [1] can be utilized. Let's assume a user has m historical picks recorded in the original collection. We

[1] User picks are the user's interaction with the system that expresses a preference or impression, like a rating, review, or feedback

consider the most recent h picks as the intended search target. The h historical picks are then grouped according to their context (for example venues in the same city, or season of visit). Each context group that has a minimum threshold of t candidates will form a separate request. In order to produce a relevance judgment for each candidate in a request, the candidate's user rating is mapped into a relevance score; i.e. if the user rated this pick positively then it is considered relevant, otherwise it is considered irrelevant.

For example, let's consider a user with 100 recorded picks in the dataset, the framework will first sort the picks by the most recently rated ones. In the case $h = 20$, the framework will take the most recent twenty picks and consider them the intended search target. Those twenty picks will then be grouped according to their context with a predetermined threshold t. If $t = 10$ this means any group with a number of picks more than or equal to ten will form a separate request. Groups with less than ten picks will be disregarded.

The personalized t-facet ranking task consists of predicting the type-based facet sub-tree to which these relevant picks belong. The remaining ratings are part of the user's history and added to the user profile in the request. To avoid creating poor user profiles, only users with a minimum of r ratings in their profile are considered for this setup. In the previous example, the user had originally 100 picks, 20 taken as search targets and 80 remain in the user profile. If $r = 100$ this means that the user has a short user profile and therefore not included in the dataset. Selecting a user with bigger r values means more historical points are available for personalization models. On the other hand, users with short user profiles require special ranking approaches to handle them, which is out of the scope of this research.

When the dataset under consideration does not provide explicit information needs, the framework generates artificial queries for each user. The queries are collected from the text associated with documents that the user has positively favored in the past (excluding the documents considered as candidates for evaluation). For this purpose, NLP methods for extracting keywords or tags can be employed to generate the top

phrases which reflect the user's interests.

The contexts and the textual query will be used as inputs to the search engine. Both the quality of the generated queries and the retrieval model affect the evaluation of the facet ranking method. The search engine must be able to retrieve the intended search target in the relevant document set, otherwise, the appropriate facet needed to reach that document could be omitted in the ranked sub-tree. On the other side, assuming that such a document is in the initial pool retrieved by the search engine, it is the objective of the t-facet ranking approach to promote it to the top of the result list.

4.1.3 Use Case 1: TREC-CS Dataset Customization

The dataset is based on TREC Contextual Suggestion (TREC-CS) track [21]. TREC-CS is a personalized point of interest (POI) recommendation task, in which participants develop systems to give a ranked list of suggestions related to a given user profile and context pair. The proposed approach solves the POI suggestion problem by ranking the types of venues as t-facets. The evaluation measures the extent to which this ranked list minimizes the user effort to reach the first relevant POI. This dataset is suitable for our task for several reasons:

1. The dataset contains POIs, which are treated as a documents. Each POI is associated with attributes and types.

2. The large well structured t-facet hierarchy (derived from Foursquare category taxonomy) demonstrates the need for t-facet ranking. The taxonomy has 5 levels and contains more than 700 categories.

3. Each request includes information about the users and their historical selections. This information is useful for personalization.

4. The existence of relevance judgments makes it possible to evaluate our approach against a well established ground-truth.

5. A number of research in literature highlighted the role played by ranking the categories of the POIs in the suggestion process [21, 64, 65].

The primary dataset is from the TREC-CS 2016 [2] and 2015 tasks [21, 74]. It has a list of POIs, user contexts, and history of selections. It also provides requests with manually assigned relevance judgments to use in an evaluation as a ground truth. On its own, this dataset is not suitable for the t-facet ranking task as it does not contain a hierarchy for venue types, nor does it contain the types of each venue. To overcome this problem, three supplementary datasets were combined to form the final dataset.

The role of the three datasets is to map TREC-CS venues to their Foursquare pages. Foursquare associates each venue with one or more categories or types. All types are organized into a well structured multilevel taxonomy. Having as many Foursquare venues linked to TREC-CS POIs as possible is important. Because type-based facet taxonomy is derived from Foursquare categories hierarchy [3].

The first complementary dataset is collated by the winners of the TREC-CS 2016 task [65]. They enriched the original dataset by crawling additional information about the POIs' from Foursquare [4]. The second dataset contained more Foursquare data crawled by the third placed team in the TREC-CS task [64]. Finally this work complements the two previous datasets by adding more Foursquare crawled venues. Finally, the three datasets are merged and integrated with the primary TREC-CS dataset. During the merger, all redundancies are removed and the described pre-processing steps in 3.2.3 are applied.

Several scoring approaches and baselines require that the resources associated with t-facets to be rated as positive, negative, or neutral. In the TREC-CS dataset the original user rating ranges from 0 to 4, with 4 being the highest rating, and 1 being the lowest one. Following TREC-CS organizers, we map 4 and 3 as positive feedback, 2 as neutral feedback, and finally 1 and 0 as negative feedback.

[2] `https://sites.google.com/site/treccontext/`
[3] https://developer.foursquare.com/docs/resources/categories, version: 20180323
[4] `https://foursquare.com/city-guide`

Query Formation

TREC-CS dataset provides requests, each request contains: 1) user related data including user age, gender and POIs rated by the user, each POI pair is associated with tags chosen by the user to describe that venue, 2) trip context that contains the trip city, duration, length, and 3) POIs' request candidates with their relevance judgments. The location, specifically the city, is used as the hard filter for the venues. The query for each request is formed from the user favorite tags. The favorite tags are those where the user rated their venues positively, they are grouped and weighted by the most common rating given to them by this user. Using the tags as input query is a common practice when dealing with TREC-CS dataset [75, 76].

Then, the submitted query, the filtered venues, in addition to the user profile and request context, are given as input to the ranking process. The query keywords are submitted in the document search engine, and they can be used again by the t-facet ranking methods to calculate query relevance features.

Venues' web pages, reviews, and categories are indexed with Solr[5], the document search engine implements BM25. As a text preprocessing step, the POI's description (content of the crawled web page) and reviews text are converted to lowercase letters and divided to sentences using Spacy sentence generator[6]. Also, numbers and special characters[7] are removed from text before being indexed with Solr. The POI fields used for search are: POI description, reviews, and list of categories. The final document ranking is boosted with rating and rating count (i.e. the POI relevance score retrieved from Solr is multiplied by the rating and rating count of the POI). The implemented search engine has NDCG value of 0.4023.

[5] `https://solr.apache.org/`

[6] The used python library: `https://spacy.io/universe/project/spacy-transformers` with BERT model en_trf_bertbaseuncased_lg

[7] Removed special characters are: $\backslash, \star, \_, \{, \}, [,], (,), >, \#, +, -, :, /, =, <, and, \wedge$.

Dataset Statistics

In this PhD's experiments only POIs linked to Foursquare are included. The first two levels of the t-facet taxonomy of the Foursquare category hierarchy are considered, these contain approximately 459 t-facets. The original TREC-CS dataset contains around 1.2 million POIs, of which 778K are linked to Foursquare resources with a known category and are therefore included in this evaluation.

The approach was evaluated on TREC-CS 2016 requests. Table 4.1 lists key statistics for the customized TREC-CS 2016 dataset, it has 58 requests and an average of 208 t-facets per request.

Item	**Statistic**
Total number of POIs	778K
Total number of taxonomy types	942
Number of taxonomy types in first 2 levels	459
Number of taxonomy levels	5
Total number of users	27 (209)
Average number of POIs rated per user	35.5 (54.1)
Total number of unique POIs rated by users	60 (4072)
Average number of rated distinct t-facets in user history	38.18
Total number of requests	57
Average number of distinct t-facets to be ranked per request	208

Table 4.1: TREC-CS 2016 Dataset Statistics after being customized using our framework (between brackets are the numbers for added data from TREC-CS 2015).

TREC-CS 2016 dataset has a lower number of requests when compared with TREC-CS 2015 [74], but a higher number of judged POIs per request which increased the number of t-facets with relevant judgments. On the other side, the TREC-CS 2015 dataset has more requests but fewer relevance judgments per request, thus fewer rated t-facets per requests. For this reason TREC-CS 2016 was more suitable for our task evaluation.

The statistics show that users in the TREC-CS 2016 dataset rated the same 60

POI, which means all users rated a small set of t-facets. In order to minimize the impact of this limitation on the ranking models, we included users and ratings from TREC-CS 2015 dataset (statistics shown in table between brackets). The reported results are using this improved user profiles. Note that this will only affect the ranking models which operate on the collaborative user feedback. It will not solve the problem for the personalization models which operate on the individual user profiles only.

4.1.4 Use Case 2: Yelp Open Dataset Customization

In this use case the framework is applied to Yelp Open Dataset (this text also refers to it using the term 'Yelp'). In order to be comparable to the TREC-CS dataset, we use it as a POI suggestion dataset. The user reviews, ratings, and POIs information are provided by the original dataset. The dataset meets the six eligibility criteria introduced earlier in section 4.1.1:

1. The published dataset contains JSON objects for the businesses (equivalent to POIs in this research), users, and reviews. Each POI object contains its attributes and types in a structured way.

2. POIs are assigned to categories derived from Yelp categories tree[8], therefore we use it as t-facet taxonomy.

3. The Yelp Open Dataset contains large number of users with their reviews and ratings. It contains long user profiles suitable for the personalization models.

4. The dataset is available to be downloaded online, no further crawling was needed.

5. It is the the same e-tourism domain like TREC-CS dataset. This domain is known in FSS literature and fits the evaluation purpose of our approach.

6. As a POI suggestion dataset, personalization plays a key role in ranking the POIs to the user.

[8]https://www.yelp.com/developers/documentation/v3/all_category_list/categories.json

To ensure rich user profiles, only users with more than 170 reviews are included ($r = 170$). We cap the user review at 1000 most recent reviews. This will help in avoiding outliers and maintaining a balanced category distribution across user profiles. For each user we take the most recent 50 reviews ($p = 50$). To create contexts, we group the reviews by their city and state. Any context with a group count of 20 or more POIs is considered a separate request ($c = 20$).

In the Yelp dataset the original user rating ranges from 5 to 1, with 5 being the best rating, and 1 being the lowest one. For the POI positive, neutral and negative mapping, the documentation from Yelp website is followed. In which the ratings 5 and 4 are considered as positive feedback, 3 is considered as neutral feedback, where finally 2 and 1 are considered as negative feedback.

Query Formation

Unfortunately the Yelp dataset does not provide textual description for the POIs, instead we index all reviews collected for each POI with Solr. Location is used as an initial filter to the document search engine. In order to build a query for each user we extract top keywords from the latest 20 reviews in the user history (excluding all candidate POIs). The keywords list was created using Rapid Automatic Keyword Extraction algorithm (Rake)[9]. Rake is a widely used method for keyword extraction in NLP literature. We compared TF-IDF generated keywords against Rake created ones, Rake keywords were meaningful and they improved NDCG results of the document search engine.

Each generated keyword was weighted by the most common rating assigned by the user to this keyword. Only keywords with the most common rating equal to 3 or more are included in the final query. This is important to reflect user preferences. Like TREC-CS, the final document ranking is boosted with rating and rating count. In addition to that, the same text preprocessing steps in TREC-CS case are followed in Yelp (described in section 4.1.3 , sub section 'Query Formation'.

[9]https://github.com/csurfer/rake-nltk

Item	TREC-CS	Yelp
Total number of POIs	778K	160K
Total number of taxonomy types	942	1,566
Number of taxonomy types in first 2 levels	459	994
Number of taxonomy levels	5	4
Total number of users	27 (209)	1,456
Average number of POIs rated per user	35.5 (54.1)	247.69
Total number of unique POIs rated by users	60 (4072)	81,163
Average number of rated t-facets in user history	38.18	135.8
Total number of requests	57	1495
Average count of t-facets to be ranked per request	208	168.14

Table 4.2: Comparing TREC-CS 2016 (between brackets are the numbers for added data from TREC-CS 2015) and Yelp Dataset Statistics after being customized using our framework.

Relevance judgments were created for each request by mapping the user rating in candidate POIs into relevance score ($score = rating - 2$), thus POIs rated 2 and 1 will be considered irrelevant. This was useful in evaluating the document ranking search engine separately. Search engine was also implemented using BM25 with NDCG value of 0.1608. We reflect on how the document ranking engine performance affects the t-facet ranking and evaluation in the next chapter (see section 5.4).

Comparing Yelp and TREC-CS Statistics

Table 4.2 shows the statistics for the two generated datasets, Yelp and TREC-CS 2016. Both datasets operate on large multilevel taxonomies. The Yelp taxonomy provides more categories which make the ranking task more challenging. The statistics also show that user profiles generated from the Yelp dataset contain a larger number of rated POIs per user, as a result we have more diverse t-facets rated by users. This provides richer data for the ranking algorithms to use in building a personalization model.

In order to experiment with longer user profiles we chose users with a large number

of historical preferences. As a result, the final customized Yelp dataset overcomes the limited user profiles problem in TREC-CS 2016 in which users rated the same 60 POIs, and each user profile contained only 30 or 60 preferences.

4.2 Personalized T-Facet Ranking Setup

4.2.1 T-Facet Scoring Methods Parameters

Probability Scoring Setup

For the probability scoring method (Prob. Scoring), the results are reported for the no background model (Model 1) and the background model (Model 2), each experimented using two coverage measures (Exact) and (Cosine). In estimating $P_r(f|q)$ (see Eq.3.3) we set $N = 1$ (for all models). Setting an N value of one will favor t-facets with high document scores at the top, which will promote the first relevant result early to the user.

Vector Space Model Scoring Setup

For the Vector Space Model scoring (VSM Scoring) , we report results for the VSM-Rocchio and VSM-Ortho methods. To choose the optimal weights for equations 3.12 and 3.11, we used hyper-parameter tuning with range of 0 to 1 and step of 0.25. In order to select the best γ value to weigh the vectors (see equation 3.8), we experimented with values ranging from 0 to 1 with a step of 0.1. The hyper-parameter tuning for the weights as well as the parameter γ was performed on all the requests of TREC-CS dataset; the best configuration was applied directly without any further tuning on the Yelp dataset requests.

In addition, the experiments include results for the vector space models using K-Nearest Neighbor (KNN) algorithm, it is called (VSM-KNN). It uses a trained KNN model for each user to predict the user's preference to a given t-facet, with $k = 1$. The model is trained using the t-facet and its positive, negative or neutral rating for each user. At query time, the input is the t-facet to be ranked, and the user's KNN model

will predict its rating. For the t-facet vector extraction we used a pre-trained BERT model[10]. It is noteworthy that at our initial experiments we also experimented with word2vec model which gave poor results. One possible explanation is that t-facet titles usually consist of two or three terms, averaging the word2vec vectors for those terms did reflect its semantics, something which the BERT vectors could achieve. In addition to that, BERT representations were re-used in other scoring methods to model the query and the t-facet using the same vector length to measure the similarity between them. It is acknowledged that this approach might benefit from other vector representation methods.

Deep Learning To Rank Scoring Setup

In results we call this scoring method DNN-LTR Scoring. For the extracted features in Group-Q the experiments used a pre-trained BERT model to compute the semantic similarity between the query and the t-facets [11]. To calculate sentiment score for the reviews TextBlob library was employed [12].

In the Yelp dataset, the features ($user\_gender$, $user\_age\_group$) were excluded from Group-P features as this information was not provided in the original dataset. In TREC-CS, the features ($user\_fans\_count$, $user\_reviews\_count$, $user\_avg\_ratings$) are also excluded from Group-P features for the same reason. Also in the Yelp dataset, the two features ($avg\_reviews\_sent$, $avg\_reviews\_sent\_weighted$) were excluded from Group-CF features. In both datasets two Group-CF features were included in the dataset, the first is avg_photos_count which is the average number of photos uploaded by the users to the documents belonging to this t-facet. Visitors might favor POIs with more uploaded photos. The second is $price\_group$, the price group to which the documents associated with this t-facet belong. It is added as a categorical feature. According to their preference, users might favor cheaper or more expensive POIs.

[10] `https://spacy.io/universe/project/spacy-transformers`, model used: en_trf_bertbaseuncased_lg

[11] `https://spacy.io/universe/project/spacy-transformers`, model used: en_trf_bertbaseuncased_lg

[12] `https://textblob.readthedocs.io/en/dev/`,version=0.16.0

The DNN model is implemented using Keras TensorFlow 2.4 Functional API . This API has the flexibility of assigning different feature group inputs to separate sub-networks. During the learning process, the whole network architecture is trained as a single DNN model. The target t-facet rank is the number of relevant POIs belonging to this t-facet, collected from the POI level relevance judgment. The model uses Mean Square Error (MSE) as a loss function. All hidden layers in the model are dense and they utilize the rectified linear activation function (ReLU), except for the last layer which employs a linear activation function.

The ranking approach and the DNN model was evaluated using ten fold cross-validation on the requests level (not on the training records level) for TREC-CS and five fold cross-validation for Yelp dataset. The learning rate used is 0.0001, and the number of epochs was set to 200 for TREC-CS and 100 for Yelp. Experimenting with more epochs did not improve the results.

Each sub-network has 3 dense layers and the number of nodes at each layer is 12, 8, and 4 respectively. The output of the four sub-network nodes are concatenated in a vector with a length of 16 elements. The vector is used as an input to the final network which predicts the t-fact score. This network also contains an additional three dense layers, each with 12, 8, and 1 (the last layer has one single node which predicts the t-facet ranking score).

We report results for the trained model using all features first (Group-All), then we report features for the trained network using each group separately (Group-P, Group-CF, and Group-Q). We also report results for the different combinations of feature groups (Group-P+CF , Group-P+Q, and Group-Q+CF).

4.2.2 Tree Construction Strategies Parameters

During our experimentation we show results for the plain list without grouping (Plain List-No Grouping), explained in section 3.4.1. This will act as a baseline to show how the effect of using different grouping strategies on the metrics compared to the original ranked facet plain list. In implementation, the system was configured to provide nine

t-facets ranked by the relevance score without any parent level grouping, if the user can not find the desired t-facet in the first page of documents results, they will press the 'More' button and the system will populate the next t-facets page.

In implementation, the document ranking engine retrieves a maximum of five matching documents per results page for the TREC-CS dataset and ten documents per page for Yelp. Using 5 results per page in the Yelp dataset resulted in many 'no relevant result found' in the facet ranking evaluation. This is one of the results of a lower NDCG value at Yelp document engine. How the quality of the document search engine impacts the facet ranking method is discussed in the next chapter section 5.4.

We compare this strategy with a Fixed level grouping strategy using two aggregation functions maximum (Fixed-Level (Max.)) and average (Fixed-Level (Avg.)), the approach explained in section 3.4.3. In order to build the tree using fixed level strategy, the system was set to return a total of nine facets per page, with three level-1 t-facets, for each three level-2 t-facets. In order to obtain comparable results, the same number of facets per page was used for the plain list baseline (Plain List-No Grouping) and plain list with grouping (Plain List with Grouping) strategies.

4.3 Baselines

With the absence of ranking methods developed for the specific case of type-based facets, a number of key facet ranking methods are collected from literature to compare the proposed approach against. They are not directly comparable to the overall approach since they do not define how the final facet tree will be constructed. They are restricted to providing a relevance score for the facets.

To provide a fair assessment during evaluation, the baselines are included as scoring methods like those suggested in the first step of our approach (see section 3.3). Their results are reported for all the implemented facet tree construction strategies (see section 3.4). Key personalized and non-personalized facet ranking methods are included in the evaluation. The baselines are listed in the following sub sections.

4.3.1 Most Frequent

This is the most common baseline in FSS literature, as discussed earlier in chapter 2. It orders the facets by counting the number of items which will appear when this facet is selected. It is a simple and quick way to rank the facet, and in many cases, it is also an effective ordering method:

$$score_{most\_frequent}(f_i, q) = freq(f_i, F_q) \tag{4.1}$$

In our experiments, this method is implemented by taking the number of the filtered venues which belong to this t-facet. In case the venue belongs to several t-facets it will be counted with each one.

4.3.2 Set-Cover

The set-cover method is introduced by Dakka et al. [47]. It also counts the venues for each t-facet, but in contrary to the previous baseline, this method makes sure each venue is only counted once with a single t-facet. In other words, it counts how many unique or new venues are covered by this t-facet.

In order to compute this score, the set-cover method follows a greedy approach, where at each step the t-facet with the maximum document count is selected first, and the documents associated with it are marked as covered. This step is then repeated for the remaining t-facets and the remaining uncovered documents.

At each loop, the t-facet with the maximum un-covered documents is selected, its documents are marked as covered and so on. Algorithm 3 explains the exact procedure followed to calculate the final t-facet score.

Algorithm 3: Set-cover greedy mutual count algorithm.

```
Input: ranked_documents
Output: scored_t_facets
t_facets_list = retrieve_categories(ranked_documents) // Returns assigned
   categories to relevant documents
scored_t_facets={} // Contains tuples of t-facets and their final score
while ranked_documents is not empty do
    <best_t-facet,best_count>=get_max_count_t-facet(
     t_facets_list,ranked_documents);      // Retrieve t-facet with highest
     count for the remaining document set.
    covered_documents=
     filter_documents(best_t-facet,ranked_documents);
    ranked_documents_set.remove( covered_documents);
    t_facets_list.remove(best_t-facet);
    scored_t_facets.append( <best_t-facet,best_count>);
end
for t-facet ∈ t_facets_list do
    scored_t_facets.append( t-facet, 0 >) // Adding the remaining t-facets
       with score equals to zero (means no added value by filtering using
       this t-facet)
end
```

4.3.3 SemFacet

This is the ranking approach followed in SemFacet faceted search system [30]. It combines three aspects of the facet ranking process. The three aspects are selectivity, diversity and depth. The exact ranking equations are described in [32].

Selectivity favors t-facets which narrow down the search results quickly, i.e. has fewer documents. It is calculated using the following equation:

$$selectivity(f_i, q) = 1 - \log_{|D_q|} freq(f_i, F_q) \tag{4.2}$$

Where diversity (also called overlap) is similar to set-cover score. It favors facets with more new documents not covered by t-facets that are ranked higher. It follows

a slightly different method to obtain normalized facet mutual count:

$$overlap(f_i, q) = \frac{1}{n + |D_q|} \times \sum_{j=1}^{n} |D_{f_i} \setminus \bigcup_{m=1, m \neq i}^{m=j} D_{f_m}| \tag{4.3}$$

Where n is the total number of facets to be ranked. The third aspect is facet depth, $depth(f_i)$, calculated as the minimum between a predefined threshold x and the current t-facet depth.

Since we apply the scoring method to end leaf nodes in a two level taxonomy, all t-facets will have the same depth, therefore this aspect will not affect the ranking process.

$$score_{SemFacet}(f_i, q) = selectivity(f_i, q) \times overlap(f_i, q) \times depth(f_i) \tag{4.4}$$

Finally, the SemFacet score is calculated by multiplying the previously mentioned aspects together, see equation 4.4. The approach targets optimizing the best ranked facet list which maximizes the score for included facets. Since it is hard to calculate all permutations of facet orders, a greedy approach is followed in selecting the t-facet with the best SemFacet score at each iteration.

4.3.4 Most Probable (Person.)

The most probable scoring method utilizes user historical ratings [14]. It is defined as the probability that the user will rate this facet positively. Its calculated by counting the t-facets for the POIs rated positively by the user, and this number is divided by total number of POIs rated by the user. This method is used as a baseline for several

papers [47].

$$score_{most\_prob\_person}(f_i, u) = \frac{rel\_freq(f_i, F_u)}{|D_u|} \tag{4.5}$$

The function $rel\_freq(f_i, F_u)$ counts how many times this t-facet was rated as relevant by the user.

4.3.5 Most Probable (Collab.)

Also suggested by Koren et al. [14], it is similar to the previous method, but includes ratings from all the users in the system, not just the current one. It counts how many times this t-facet was rated positively by all system users divided by the number of POIs rated in the system.

$$score_{most\_prob\_collab}(f_i, U) = \frac{rel\_freq(f_i, F_U)}{|D_U|} \tag{4.6}$$

The function $rel\_freq(f_i, F_U)$ counts how many times this t-facet was rated as relevant by all user in the system.

4.3.6 MF-SVM

The method implements matrix factorization using Support Vector Machine (SVM). According to the authors, SVM is the best performing MF technique for facet ranking [42]. The ratings matrix is built by adding the users and their t-facet ratings in the range from 1 to 4. T-facet ratings are collected from their POIs' ratings.

It is usually the case that the same facet is rated several times as the user rates several POIs, therefore it has multiple ratings from the same user. In this case, this method takes the median of the t-facet rating values. In the implementation, the Surprise [13] python library is used to train and predict the t-facet rating [77].

[13] `http://surpriselib.com/`

4.3.7 TF-IDF Similarity

This baseline method is inspired by [31] and [40]. It was originally developed to highlight interesting facets to the user in a graph visualization tool. During this process the method generates a personalized facet weight which reflects its relevance to the user. Hence it is comparable to other facet ranking methods.

In order to compute this score, the algorithm starts by building bag of words profile for each user and each t-facet. The user profile contains the top n terms collected from their rated documents or reviews, determined using the term's TF-IDF score. In the same way, for each t-facet a words profile is built from the documents belonging to this t-facet using TF-IDF score.

Finally, the t-facet score is the similarity between the user profile and the t-facet profile. Only words tagged as 'Nouns' are considered by this approach, stop words and verbs are excluded. The nouns are tagged using NLTK[14] python position tagger.

4.4 Evaluation Strategy

4.4.1 Overview

The evaluation assesses whether the personalized facet ranking makes it easier for the user to reach the target resource. The simulated users approach proposed by Koren et al. [14] is followed in evaluating the proposed method. The method is widely adopted across FSS evaluation [1, 3, 17, 56].

The Koren et al. approach counts how many clicks the user had to make in order for the intended resource to appear in the top search results. The simulation uses several navigation strategies to reach the target group. The improvement will be measured by comparing personalized and non-personalized baseline approaches.

This simulation approach, combined with metrics provided by INEX 2011 Faceted Search task [1, 67], will serve as the core of our evaluation method. Metrics suggested

[14] `https://www.nltk.org/`

by the task organizers are based on the *first-match* user interaction model proposed by Koren et al. In this work, the facet evaluation metrics are adjusted to fit the special type-based facet case, in section 4.4.3, this adaptation is illustrated with a flow chart to explain how the metrics will be calculated. The approach is also adjusted to treat facets as levels (level-1 and level-2 in our case) rather than considering them facet and facet-values.

4.4.2 Assumptions

The first assumption is set by Koren et al., they assumed that the user can recognize their intended search target and has the knowledge of which type-based facet can be selected to lead to the intended search target.

The user fulfills her/his search need when she/he finds the relevant search target. To achieve this, the approach followed by INEX 2011 data centric task was followed. In which the searcher needs are fulfilled when the first relevant result is found.

The adopted evaluation strategy also assumes single facet selection model. The t-facet list is populated once at the beginning of the session and that population does not change after each t-facet selection.

While this assumption is suitable for t-facets, it might not be suitable for other facet types, like in the case of p-facets. For property-based facets, indeed, the users selection of one or multiple facets may change the order of presentation of the others.

4.4.3 Users Simulated Interaction Model

We adjust the interaction model proposed by the INEX 2011 Data Centric task organizers [1]. The adjusted model simulates user interaction with a hierarchical tree of facets rather than a list of facets and their values.

Figure 4.1 illustrates the simulation process and how the metrics are calculated at each step. The process assumes that the user knows the target resources tr and their associated t-facets. INEX task simulation uses the *First Match* method proposed by

Figure 4.1: Adjusted simulated user interaction model

Koren et al. [14]. In this method, the users select the first matching t-facet as soon as they see it, regardless of whether this t-facet will promote the relevant document higher or not.

Koren et al. also proposed MYOPIC and Stochastic interaction models. In the MYOPIC interaction model, the user selects the relevant facet which will narrow down the results quickly. For example if the intended document has multiple types, the MYOPIC user will search for the associated t-facet with less results count. Whereas the Stochastic user will select any random relevant t-facet. In our early trials we experimented with three methods and found that First Match and MYOPIC users give the same indicators about ranking methods performance, so we followed INEX method and adopted only First Match method.

The simulated procedure kicks off after the first page of results and the first page of t-facet tree are populated. The user scans the first page of documents first to look for the target resource, if not found the user starts scanning the t-facet tree for a relevant leaf-node t-facet.

If the t-facet node is not found in the first t-facets page, the user starts to clicks on the 'More' link till she/he reaches the desired result. During this interaction process, the system calculated the Number of Actions (# Actions) and F-Scan metrics. We explain both in detail in the coming sections.

4.4.4 Metric 1: Number of Actions (# Actions)

This metric counts how many clicks the user had to perform in order to fulfill her/his search needs. This metric was originally suggested by Koren et al. [14] and it follows the simulated user interaction model described earlier. For example, in figure 3.7 each click on 'More Cat D' or 'More Categories' will be considered an action. Selecting the document is also considered an action.

The minimum value for this metric is one, which means that the user could find the intended search target in the first page of results and did not have to use the

t-facets to filter the results at all. The number of actions metric will increase as the user navigates between t-facet pages to search for the intended category.

In the results, the average number of actions are reported for all requests. The selected baselines are compared against scoring methods of the proposed approach. The t-facet page size is set to three per each level and a total of nine t-facets per page. This small number is chosen to better evaluate the t-facet ordering and penalize the poorly ranked t-facets. In real life cases, the selection of this number is governed by other usability aspects.

4.4.5 Metric 2: Facet Interaction Cost (F-Scan)

Interaction cost is defined by the number of t-facets plus the number of documents the user had to scan before fulfilling her/his search needs. This metric was originally introduced by Kashyap et al. [53] and used in INEX 2011 Data Centric Task [1]. It uses the same evaluation assumptions and user interaction model proposed by Koren et al. [14]. Furthermore, when adopting this metric in INEX 2011 Data Centric Task, it was assumed that the cost of scanning the facet is the same cost as scanning a document, both equal to one. This assumption is followed during this research experimentation.

It is usually compared against raw cost for a non-faceted search system which is calculated by counting how many documents the user had to scan before finding the first relevant result, it is similar to the Reciprocal Rank in traditional IR metrics. This work's experimental results compare the average faceted cost for all requests between several baselines. The less the cost, the better the system.

4.4.6 Metric 3: F-NDCG

The third and final metric included in the evaluations of the proposed approach is called F-NDCG [67]. It inherits the merit of the widely used IR metric NDCG but is tweaked for the case where the relevance judgments on the facets level are unknown, but they are known on the document level. F-NDCG is built on the idea that relevance

is transferred from the documents to the facets they are associated with.

It was developed specifically for INEX 2011 Data Centric Task evaluation [1]. In this work, the 'F' is added before NDCG to differentiate the facet ranking metric from the document level NDCG one. The metric favors highly ranked t-facets which cover more new relevant documents. Like the traditional NDCG it can be calculated over the whole result set or on the first n t-facets F-NDCG@n. It is calculated using equation 4.7:

$$\text{F-NDCG(D,F,q)} = \frac{DCG(D,F,q)}{IDCG(D,F,q)} \tag{4.7}$$

Like the traditional IR metric, the facet ranking version normalizes the discounted cumulative gain for each t-facet by its ideal discounted cumulative gain. The discounted cumulative gain (DCG) is calculated using the following formula:

$$DCG(D,F,q) = \sum_{i=1}^{min(n,|F_q|)} \frac{Gain(D,F,f_i,q)}{\log_2(i+1)} \tag{4.8}$$

Where n is the position the F-NDCG is calculated at. Equation 4.8 describes regular discounted cumulative gain, but the gain definition in facets case is different, It counts how many 'new' relevant documents are covered by this t-facet, see equation 4.9.

$$Gain(D,F,f_i,q) = |D_{f_i} \setminus \bigcup_{j=1}^{i-1} D_{f_j}| \tag{4.9}$$

Likewise, the ideal discounted cumulative gain is calculated in the following equation:

$$IDCG(D, F, q) = \sum_{i=1}^{min(n,|F_q|)} \frac{IGain(D, F, f_i, q)}{\log_2(i+1)} \tag{4.10}$$

Where the ideal gain (see eq. 4.11) in this case states that any facet can cover at most p new relevant documents. Where p is the maximum number of relevant documents, which can be covered by any t-facet.

$$IGain(D, f_i, q) = max(0, min(p, |D_q| - (i-1) \times p)) \tag{4.11}$$

Finally, in results, average F-NDCG values for all requests or queries is reported for the compared systems. The F-NDCG metric authors also proposed a recursive version of the metric, which considers different user interaction models in which facet-values change after each new action by the user. Since the focus of this research is type-based facets and they are populated only once at the beginning of the interaction process, this F-NDCG version is more suitable to the targeted research task than the recursive one.

4.4.7 Metrics Interpretation

In order to answer the research question targeted by this thesis (see section 1.3), the number of actions is considered as the main evaluation metric to be used in ranking algorithms comparison. The metric is less affected by the quality of the underlying document ranker, where the facet interaction cost metric (F-Scan) includes the document page scan cost to the metric which might overshadow the performance of the t-facet ranking. It also assigns the same cost to both facet scan and document

scan effort. For these reasons it is considered a complementary metric to the number of actions one.

In contrary to the first two metrics, the F-NDCG metric does not directly measure the effort undertaken by the user to fulfill search needs. However, F-NDCG calculation is useful in assessing the utility for the overall ranked t-facet list. It is also a useful metric in deciding the target label for the LTR scoring approach.

In general, combining the three metrics and comparing them across several baselines and with different datasets will give useful insights about the behavior of the ranking methods. Other metrics mentioned in literature like RMSE to measure if the algorithm could predict the correct facet would not be a good fit for this research task [42]. It will convert the task into a category prediction, while it might be useful for one target system, it does not scale to different system types and it does not explore the characteristics of faceted search, nor adhere to the user interaction model.

Chapter 5

Results and Discussion

In this chapter the results of the final round of experimentation are presented. As we experiment with two datasets, we are interested in studying how the proposed approach behaves from different angles. With this in mind, we experimented with different scoring methods, tree building strategies and produced results for three different metrics for each run.

During the evaluation we will focus on the number of actions (#Actions) as the main metric (explained in section 4.4.4), since this metric measures the users effort to reach the intended target as discussed previously in section 4.4.7. In addition to that, the metric counts actions performed on the ranked facet list, therefore it is less affected by other factors (like the quality of the document ranking engine). The #Actions metric will help answer the research question posed in the thesis: *Can personalizing type-based facet ranking according to user historical feedback minimize the effort needed by users to fulfill their search needs?* (see chapter 1, section 1.3). Although we will also report the other metrics because in specific contexts we want to use them to draw some specific conclusion or to look as some specific aspects/behaviour of the proposed methods. The full results including F-Scan and F-NDCG results are computed and included in the Appendix A1.

Additionally, throughout the results we list only one tree building approach: Plain List-with Grouping, described in 3.4.2. It was selected as it consistently provided the best results across the scoring methods. Tables containing the complete results for other tree building approaches can be found in the Appendix A1 and referenced in discussions as needed. Results are reported on both datasets, TREC-CS and Yelp, and are described in sections 4.1.3 and 4.1.4 respectively. All reported statistics in this chapter are rounded to three decimal places. Moreover, a statistical significance test was performed on the proposed methods using the 'Most Frequent' method as a baseline and results are reported in all tables. The reported P values are calculated using one-tailed paired t-test to measure improvement for the #Actions metric over the baseline.

Key conclusions from all results (including those from the Appendix) are included at the end of this chapter. Before diving into the experimental results, the following section sheds some light on the effort spent on the design and development of the chosen experimental setup and how different implementation challenges were tackled during this process.

5.1 Implementation Rounds

While conducting this research, an iterative approach to design and implement the experimental setup was followed. This was an essential journey for the development and articulation of the final approach and evaluation strategy presented in this thesis.

At the earlier stages of this research, several components of the faceted search systems were explored before narrowing down to the facet ranking component. Facet ranking was an interesting part because this is where the personalization happens. After several initial experiments on attribute ranking and triple scoring, it was obvious that type-based facets would benefit from a ranking method especially designed for it.

As the experiments progressed, the need for a tree construction step was realized.

A separate additional step to display the final tree was included. The main idea was not to find the best construction strategy but rather highlight how different strategies will affect the evaluation metrics [1]. Grouping the t-facets by their parents sounds intuitive, however several strategies can be followed to do this grouping, each will result in a different tree and therefore will affect the evaluation metrics. In the results, we focus on and study this part.

One of the key challenges faced during the development of the system is the lack of a benchmarked dataset. The process of finding an appropriate domain and an available dataset took a considerable amount of time. When not found, that led to the formation of the proposed dataset creation framework, where the criteria were outlined and the required dataset structure needed to be formalized.

The TREC-CS dataset was the first dataset to be customised to fit the purpose of evaluating this approach. The original dataset was complemented by two other datasets [64, 65]. They were combined to obtain the categories of the POIs as this data was not provided in the original dataset. In addition to that, as part of this research a crawler was developed to collect more POI information from the Foursquare website to be added to the dataset.

Several baselines were implemented and tested on the TREC-CS dataset. By investigating the preliminary results obtained, the limitations of short user profiles became apparent. Some findings for the personalization models were inconclusive, to confirm them another dataset was needed. This led to the adoption of the Yelp dataset, which confirmed the findings as it contained richer and more diverse user profiles.

For both the TREC-CS and Yelp datasets, they were indexed into two data stores, the first is a No SQL MongoDB database containing the POIs, attributes, crawled text, and reviews. This database was utilized for the straightforward retrieval using a POI identifier. In addition to that all POIs' textual content was also indexed

[1] It is acknowledged that there are several UI considerations that contribute to facet tree construction. This work focuses on how this step impacts the evaluation metrics.

for keyword search using Solr, which acted as the document search engine for our approach.

From the evaluation end, as existing FSSs follow a variety of evaluation strategies, at the early stage of this research, we implemented several versions of simulation based approaches which gave either inconsistent or confusing results. The main problem was how to derive the ground-truth on the facets level, given the lack of relevance judgments for the facets themselves.

It was a reasonable alternative to assume that the relevance of the facets can be driven from the relevance of the documents associated with it. Following the citations of the well-known Koren et al. [14] method, the INEX 2011 Data Centric task was found. Which pursued this alternative and provided the evaluation setup focused on the facet ranking evaluation. They also introduced a number of metrics to measure the performance of the ranking methods as well as the user interaction model. They also demonstrated how it can be applied to the task's dataset. This evaluation protocol provided a sound methodology to adapt and implement into our own evaluation.

However, the evaluation framework was intended for the generic facet ranking methods, it had to be further adapted to the hierarchical t-facet case. Because of the novelty of this research area, no existing implementations for the user simulation model nor the metrics calculations were found, it had to be implemented from scratch and tested thoroughly to make sure the calculations are correct. At this point the entire code base of this work was refactored to incorporate the new evaluation approach.

As part of this distraction, a whole framework was developed for the the facet ranking baselines, hyper-parameter tuning, feature generation, dataset statistics calculation, document ranking search interface, data loading, NLP preprocessing, and of course the proposed ranking functions and requests. The implemented system was built to be easily modified and flexible enough to repeat the experiments as many times as needed.

5.2 Approach Evaluation Results

This section covers the results of the proposed scoring methods introduced in section 3.3 using the experimental setup described in section 4.2. It reports and discusses the results of each scoring first, and then the following section 5.3 compares the performance of the best performing models with the baselines.

5.2.1 Probabilistic Scoring Method (Prob. Scoring)

The results in Table 5.1 show that the no background Model-1 consistently outperforms the background Model-2 across # Action, F-Scan, and F-NDCG metrics (see appendix for F-Scan and F-NDCG values, Tables A1.1 and A1.2)). This is true regardless the coverage method used (introduced in section 3.3.1), and for both datasets across different tree construction strategies (see Appendix A1, Tables A1.1 and A1.2). The key difference between both models is that Model 1 estimates the t-facet score based on the individual user profile, whereas Model 2, combines the individual user profile with background data collected from all users in the system. This indicates that the individual preferences alone had a stronger influence than being combined with other user preferences as background data in both datasets.

Scoring Method	TREC-CS			Yelp		
	Mean	95% CI	p-value	Mean	95% CI	p-value
Model 1 - Exact	1.474	(1.13, 1.5)	0.940	2.252	(2.21, 2.4)	0.545
Model 1 - Cosine	**1.316**	**(1.13, 1.5)**	0.392	**2.147**	**(2.1, 2.29)**	0.069
Model 2 - Exact	1.509	(1.2, 1.82)	0.935	2.609	(2.54, 2.75)	1.000
Model 2 - Cosine	1.579	(1.2, 1.96)	0.950	2.589	(2.55, 2.79)	0.999

Table 5.1: Average #Actions with 95% confidence intervals (CI) for probabilistic scoring models using Plain Tree with Grouping strategy (TREC-CS sample size=57, Yelp sample size= 1,495).

It can also be observed in Table 5.1 that *Cosine* similarity implementation gives better results in Model-1. One possible explanation that using cosine similarity aided the score generation for new unseen, thus unrated, t-facets, where the strict *Exact* matching approach fails to handle such cases since it assigns 0 score if the user never

rated that category before.

For the best performing scorer (Model 1-Cosine), the Plain Tree with Grouping strategy produced minimum #Actions value results across different tree construction strategies (see Tables A1.1 and A1.2 for complete results). Two factors caused this: 1) The tree construction strategy maintained the top scored level-2 facet at the top of the final tree. 2) In estimating $P_r(f|q)$ (see Eq.3.3) we set $N = 1$ (for all models) to favor t-facets that promote the first relevant result early to the user, which in turn effectively minimized the user effort as shown in the results. When experimenting with higher N values, all metrics were negatively impacted.

5.2.2 Vector Space Models Scoring Method (VSM Scoring)

Amongst the three Vector Space Models included in the experiments, the VSM-Rocchio method outperformed both VSM-Ortho and VSM-KNN methods. The normalization step by the Gram-Schmidt process used in Ortho-BERT neutralises the personalized negative profile weight, which negatively affects the results. Using the traditional KNN clustering approach was not useful in this task, as experimenting with different k values did not improve its results.

Scoring Method	TREC-CS			Yelp		
	Mean	95% CI	p-value	Mean	95% CI	p-value
VSM-Rocchio	**1.263**	(1.14, 1.39)	0.0364*	**1.997**	(1.96, 2.13)	0.0003*
VSM-Ortho	1.333	(1.16, 1.5)	0.500	2.264	(2.22, 2.41)	0.621
VSM-KNN	1.561	(1.26, 1.86)	0.993	2.862	(2.82, 3.03)	1.00

Table 5.2: Average #Actions with 95% confidence intervals (CI) for VSM scoring method using Plain Tree with Grouping strategy (TREC-CS sample size=57, Yelp sample size= 1,495), * denotes statistically significant result at p-value$<$0.05.

VSM-Rocchio achieved the best results with the weight of neutral user vector $\alpha = 0$, where β is the weight of the positive user vector $\beta \in \{0.25, 0.5, 0.75, 1\}$, and $\lambda = 0$ which is the weight for the negative user vector. These results reflect that the positive vector profile is the key component in the formula. $\gamma = 0$, which indicates that the individual person probability is favoured over the global probability in the

t-facets weighting (See Eq. 3.8).

The proposed VSM approach has some advantages. It provides a light yet effective personalized ranking. The user profile vectors can be pre-computed and stored offline, reducing the computation at retrieval time of the cosine distance between the query t-facets and the user profile.

This can be further optimized by pre-computing a distance matrix for the user and all t-facets in the FSS. User profiles on the other hand, can be updated offline as the user rates new venues. This is indeed useful, as the t-facet ranking step occurs after the document ranking. Optimizing this step will reduce the total time needed by FSS to populate its final results page.

5.2.3 Deep Learning To Rank Scoring Method (DNN-LTR Scoring)

Table 5.3 summarizes the DNN-LTR scoring method results for the two datasets included in our experiments. The table reports the DNN model results using different combinations of feature groups. This is important in understanding the contribution of each group to the t-facet ranking score. We report the results of our DNN approach using all groups of features denoted by (*Group-All*), then for each group used individually: the individual personalization features group (*Group-P*), and the collaborative filtering group of features (*Group-CF*), and the query related features group (*Group-Q*). In addition to that, we also report results for the combination of each group of features, for the features of Group-Q plus the features of Group CF are reported as (*Group-CF+Q*), and the same for the combination of Group-CF and Group-P denoted by (*Group-CF+P*), and final Group-Q with Group-P as (*Group-Q+P*).

In the results table 5.3, we can see that Group-Q and CF, separately and combined, achieved the minimum number of actions in the TREC-CS dataset. Group-Q good performance is due to the strong features it contains. These features are extracted from the document level ranks. The case is not the same in the Yelp dataset, because the performance of the document ranking engine implemented in TREC-CS

Scoring Method	TREC-CS			Yelp		
	Mean	95% CI	p-value	Mean	95% CI	p-value
Group-All	1.316	(1.16, 1.48)	0.349	2.127	(2.09, 2.26)	0.027*
Group-P	1.316	(1.15, 1.48)	0.368	**2.108**	**(2.08, 2.23)**	0.011*
Group-CF	**1.281**	**(1.14, 1.42)**	0.245	2.164	(2.13, 2.3)	0.095
Group-Q	1.298	(1.14, 1.46)	0.102	2.158	(2.13, 2.29)	0.077
Group-P+CF	1.333	(1.16, 1.5)	0.500	2.129	(2.1, 2.25)	0.026*
Group-Q+CF	**1.281**	**(1.14, 1.42)**	0.102	2.175	(2.14, 2.31)	0.132
Group-P+Q	1.439	(1.21, 1.66)	0.962	2.134	(2.1, 2.26)	0.035*

Table 5.3: Average #Actions with 95% confidence intervals (CI) for DNN-LTR scoring method using Plain List with Grouping strategy (TREC-CS sample size=57, Yelp sample size=1,495), * denotes statistically significant result at p-value$<$0.05.

is better than the Yelp document ranker (NDCG value of 0.4023 in TREC-CS versus NDCG value of 0.1608 in Yelp). The reason behind this, is that in TREC-CS case the query was constructed from manually curated user tags so they retrieved more relevant documents. A better document ranker means that info-gain features are more useful for ranking. And a good query means that also the query similarity features are meaningful. On the other hand, Group-Q behaved poorly on Yelp as in that case the query was automatically created from keywords extracted from previous users' reviews. This results in noisy keywords and therefore affects the document ranking performance (NDCG=0.1608). As a result, Group-Q features are less useful in training the ranking model. This effect should be considered during the process of selecting an appropriate t-facet scoring method.

By investigating the DNN-LTR approach results on the TREC-CS dataset across all tree construction strategies (see table A1.5), considering the individual groups of features, the performance of the three groups seems comparable, it is fair to assume that when combined, the groups contribute equally to the results of the Group-All model. In general, we can see that this scoring method gives better results with respect to F-NDCG metric when compared to the VSM and the probabilistic scoring methods, although when we inspect the #Actions metric the results for the TREC-CS are inconclusive.

We can also observe that there is no winner group of features. Groups results are inconsistent across metrics and tree construction strategies (see table A1.5). This might be due to the relatively small number of training records (60 requests in TREC-CS resulting in around 5500 training records). This affected the DNN model's learning, even though all groups of DNN-LTR approaches achieved best the F-NDCG values when compared to other scoring methods.

Turning to the Yelp dataset results in table 5.3, we found that DNN-LTR models trained either with all the features included (Group-All) or with only personalization features, when evaluated consistently outperformed other groups regardless of the tree construction strategy used. This might be explained by several factors. First, DNN models on Yelp are trained on more than 1400 requests with 290,000 record. Second, the personalization features (Group-P) are richer as Yelp user profiles contain at least 150 historical POIs. Where users in TREC-CS have limited profiles of either 30 or 60 ranking POIs, however, Group-P was able to achieve the best F-NDCG values in TREC-CS dataset, although the model failed at minimizing the number of actions metric. Finally overall DNN-LTR models results in yelp, we can see that they also achieved the best F-NDCG values when compared to the VSM and Probability Scoring methods.

5.3 Comparing Approach Results with Baselines

This section compares the approach evaluation results against several personalized and non-personalized baselines methods previously introduced in section 4.3. Tables 5.4 and 5.5 compare the proposed approach performance against each other and four non-personalized baseline methods for TREC-CS and Yelp datasets. The first group of scoring methods contain the methods proposed by this thesis; for each scoring method the table only includes its best performing version. In the probability scoring, results are reported for the Model 1 + Cosine model. Meanwhile for VSM Scoring the VSM-Rocchio results are used. Lastly for the DNN-LTR scoring method, the results are added for the Group-Q for the TREC-CS dataset and Group-P for the

Method Type	Scoring Method	#Actions			F-NDCG	
		Mean	95% CI	p-value	Mean	95% CI
Our methods	Prob. Scoring	1.316	(1.13, 1.5)	0.392	0.156	(0.133, 0.18)
	VSM Scoring	**1.263**	(1.14, 1.39)	0.036*	0.117	(0.098, 0.136)
	DNN-LTR Scoring	1.293	(1.14, 1.46)	0.245	**0.242**	(0.212, 0.271)
Non-personalized	Most Frequent	1.333	(1.16, 1.5)	-	0.216	(0.188, 0.245)
	Set-Cover	1.368	(1.18, 1.56)	0.740	0.201	(0.175, 0.228)
	Alphabetical	1.438	(1.2, 1.67)	0.934	0.089	(0.069, 0.11)
	SemFacet	1.631	(1.3, 1.96)	0.999	0.013	(0.008, 0.02)
Personalized	MF-SVM	1.316	(1.15, 1.48)	0.368	0.121	(0.098, 0.144)
	Most Prob. (Person)	1.491	(1.2, 1.78)	0.936	0.222	(0.192, 0.251)
	Most Prob. (Collab)	1.368	(1.18, 1.56)	0.740	0.226	(0.195, 0.258)
	TF.IDF	1.298	(1.15, 1.45)	0.210	0.169	(0.144, 0.193)

Table 5.4: Comparing #Actions and F-NDCG for the proposed approaches and baselines using Plain List with Grouping strategy for TREC-CS dataset (sample size=57), * denotes statistically significant result at p-value$<$0.05.

Yelp dataset.

Following that, the tables present the results for the implemented baselines (introduced previously in section 4.3). The non-personalized methods are: Most Frequent, Set-Cover, Alphabetical, and SemFacet. The personalized methods are: MF-SVM, Most Prob. (Person), Most Prob. (Collab) and TF-IDF similarity. The methods are grouped according to their type for better readability. Since none of the existing methods handles the hierarchical nature of the t-facets, we use them as scoring methods. Their results are reported using Plain List with Grouping strategy. Moreover, histograms for the #Actions metric are included in table A1.9 in Appendix A1.

Considering the number of actions metric, the VSM scoring method outperforms other scoring methods and the baselines. The DNN-based models came second with the benefit of maximized F-NDCG values across both datasets. They outperform all the baselines in TREC-CS and Yelp datasets. This reflects how the DNN-based models have the ability to generate trees with a collection of relevant t-facets, rather than producing a tree with a single relevant t-facet at its top, as in the case of the

Method Type	Scoring Method	#Actions			F-NDCG	
		Mean	95% CI	p-value	Mean	95% CI
Our methods	Prob. Scoring	2.147	(2.1, 2.29)	0.069	0.049	(0.047, 0.053)
	VSM Scoring	**1.997**	(1.96, 2.13)	0.0003*	0.052	(0.051, 0.056)
	DNN-LTR Scoring	2.108	(2.08, 2.23)	0.011*	**0.076**	(0.073, 0.082)
Non-personalized	Most Frequent	2.244	(2.2, 2.39)	-	0.071	(0.069, 0.077)
	Set-Cover	2.312	(2.27, 2.46)	0.855	0.067	(0.065, 0.072)
	Alphabetical	3.642	(3.54, 3.91)	1.00	0.006	(0.005, 0.007)
	SemFacet	3.319	(3.26, 3.53)	1.00	0.004	(0.004, 0.005)
Personalized	MF-SVM	3.342	(3.26, 3.57)	1.00	0.006	(0.006, 0.007)
	Most Prob. (Person)	2.117	(2.08, 2.25)	0.019*	0.074	(0.072, 0.080)
	Most Prob. (Collab)	2.155	(2.12, 2.29)	0.073	0.068	(0.066, 0.073)
	TF.IDF	2.388	(2.35, 2.54)	0.985	0.044	(0.042, 0.047)

Table 5.5: Comparing #Actions and F-NDCG for the proposed approaches and baselines using Plain List with Grouping strategy for Yelp dataset (sample size=1,495), * denotes statistically significant result at p-value$<$0.05.

VSM Scoring method. While the VSM scoring method has the ability to predict the top t-facet effectively, it failed to produce a whole relevant t-facet tree to the user.

5.3.1 Non-personalized Baselines

With respect to the non-personalized baselines, the most commonly used method in FSS literature (Most Frequent), which remains a very strong baseline. From both tables 5.5 and 5.4 it can be seen that it is the best performer among the non-personalized baselines. The key advantage of this method is that counts can be easily computed and it provides reasonable results. This might be suitable for the cases where the document ranking engine provides high precision results, otherwise the simple count will be affected with noise propagated from the document ranking step. Besides, as it promotes popular facets among results the method will fail in cases where users are picky and their preferences are not popular.

The same for the Set-Cover method, it is highly dependable on the quality of the underlying document ranking engine. On its own the greedy version of Set-Cover

did not provide any improvement over the simple Most Frequent method. Other optimized versions of the method might lead to better results but they will also include heavier computations.

Although adopted by several FSSs, ranking the t-facets alphabetically fails to recommend useful t-facets to the user over both datasets. Similarly SemFacet consistently performed the worst across scoring methods and tree construction strategies in both datasets. In addition to performing a very heavy computation, the selectivity and overlap factors neither reflect t-facet general importance nor its relevance to the user. This method might be developed to be used in specific scenarios not for precision-oriented FSS.

5.3.2 Personalized Baselines

Now turning to the personalized methods. For the MF-SVM baseline the results were below expectations since matrix factorization methods are famous for their ability to capture user interests. In order to give it as much users as possible we trained the model using both the TREC-CS 2015 and 2016 users and their historical picks. This did not improve its results. The model also performed poorly for the Yelp dataset; this indicates that the adoption of the average of document ratings as a t-facet target rank is a poor heuristic to rate t-facets when many diverging ratings need to be aggregated.

Moving to the TF-IDF method, it performed well on the TREC-CS dataset, where the user profiles are built from their historical preferences, that is the description of the POI previously rated positive by the user. Matching this with t-facet profile produced best results amongst personalized baselines across all tree-construction strategies (see tables A1.7 and A1.8). The TF-IDF method has an advantage over the other baselines given that it caters for individual user preferences while handling new and unseen t-facets.

On the disadvantages side, the method failed at maximizing F-NDCG metric, which implies it promoted a number of irrelevant t-facets to the first t-facets page. In

addition to that, building and maintaining updated user profiles will require complex NLP computation and storage to index all POI TF-IDF terms for the t-facet ranking approach. The same is true for building and updating the t-facet TF-IDF profiles. In addition to that the method relies on the existence of good quality descriptive content for each POI, which might not be the case in real-life datasets. Point in case is the TF-IDF method behavior on the Yelp dataset, where the model performance dropped as the dataset lacks textual description for the POI, when the profile is not substituted by the available reviews and tips for each POI, it fails to capture user preferences adequately, therefore produced poor rankings. In contrary, all our proposed scoring methods do not require heavy NLP processing in building and updating user profiles and t-facet profiles. The profiles are created and updated using straightforward calculations.

Most probable (collab) and (person) methods achieved highest F-NDCG values across personalized baselines. The Most probable (Collab) performed better on the TREC-CS dataset. The reason being that the approach favors popular t-facets, which worked well given the skewed t-facet probabilities in the dataset. This is because in the TREC-CS 2016 all users rated the same 60 POIs, as a result a limited set of t-facets are rated by all users. Turning to the most probable (person) method in the TREC-CS, it performed worse than the collab. method in the #Actions metric, this is due to the limited profiles in the TREC-CS 2016 dataset as each user rated either 30 or 60 POIs.

Yelp dataset overcomes the limited user profiles problem in TREC-CS 2016, this affected category distribution of the rated t-facets. The users in Yelp rated more POIs, therefore more t-facets are also rated creating a more diversified and realistic category distribution for the t-facets. The effect of the quality of personal profiles is more apparent in the Most Probable (Person.) performance. As the ranking method mainly depends on the users historical ratings. The approach improved in both metrics in the Yelp dataset as well as all scoring methods based on individual user profiles. In an attempt to minimize the limited profile issues we included users and ratings from the TREC-CS 2015 dataset. The reported results are using these

improved user profiles, this however did not resolve the issue for the personalized baselines.

Besides, both methods have the disadvantage of not handling new unseen t-facets, they also fail for users with unpopular preferences. For example, the Most probable (collab) method will result in a t-facet tree ordered by the t-facets popularity in the dataset. Like the Most Frequent method, this fails to take account of users with unpopular preferences. Our proposed approaches on the other hand, handle both cases effectively.

In order to investigate the significance of the produced results, a hypothesis testing is run to determine whether the proposed approach produced significant improvement over non-personalized methods or not. The null hypothesis, H_0, states that personalized t-facet ranking methods do not improve the number of actions metric. Based on results, the best performing non-personalized scoring method across both datasets and all tree construction strategies is the Most Frequent method. For this reason it was considered the baseline for null hypothesis. The P value is calculated using one-tailed paired t-test to measure improvement for the #Actions metric over this method. A p-value of 0.05 is used as the cutoff for significance level in the performed statistical tests. This value is commonly used in IR literature.

The first alternative hypothesis H_1: is that the best existing personalized baseline has introduced improvement over the best non-personalized ranking method. This p-value was computed for the TF-IDF approach on TREC-CS dataset (for exact p-values see table 5.6), and Most Probable (Person and Collab) for Yelp dataset across tree construction strategies(for exact p-values see table 5.7). Although these personalization baselines achieved competitive results there was no sufficient evidence that they improved the number of actions metric. This is true for the TREC-CS dataset as well as the Yelp dataset.

Another alternative hypothesis was also tested H_2: it states that our proposed personalized t-facet ranking approach improved the number of actions metric, when compared against the best non-personalized baseline. In the Yelp dataset, DNN-LTR

Group-P and Group-All both achieved statistically significant results for the Plain test with grouping strategy, Fixed-Level (Max) and Fixed-Level (Avg) (for exact p-values see table 5.7).

In the TREC-CS dataset, only requests where the t-facets were used are considered (#Actions> 1). Because the good quality of the document ranking engine most of the relevant targets are found in the first result page, therefore the user did not use the t-facet list at all. The number of requests with number of actions more than 1 is only 14 requests. Keeping the requests with # of actions equal 1 affected the mean and standard deviation of the metric. In order to focus on the improvement introduced by the ranked t-facets, the requests with number of action equal to one was excluded from the statistical test. This issue does not exist in the case of Yelp dataset as the number of requests with single action was very small. VSM Scoring using Rocchio was found to significantly improve the results at using Plain List with Grouping strategy (for p-value see table 5.4), where DNN-LTR Group-P+CF was found to significantly improve the number of actions metric with Fixed Level (Avg) tree construction method (p-value = 0.036). Based on the outcomes from both datasets, it is found that there is sufficient evidence to conclude that the proposed personalized t-facet ranking approach improved the number of actions metric, when compared against the best non-personalized baseline.

5.3.3 Impact of Using Different Tree Construction Strategies

In order to further understand the effect of using different tree construction strategies, Tables 5.6 and 5.7 report results of the proposed approaches and the best performing baselines using the four tree construction strategies suggested in section 3.4. The strategies are: Fixed Level (Max) , Fixed Level (Avg), Plain List with Grouping strategy, and Plain List-No Grouping. The Plain List-No Grouping is the equivalent of a plain list of t-facets not organized in a tree. This strategy acts as a baseline to the study of the impact of using different tree construction strategies and how they affect the evaluation metrics. The best scoring method results are highlighted in bold for each strategy in table 5.6 for TREC-CS dataset and table 5.7 for yelp dataset.

Tree Strategy	Scoring Method	#Actions			F-NDCG	
		Mean	95% CI	p-value	Mean	95% CI
Fixed Level (Max)	Prob. Scoring	1.333	(1.12, 1.54)	0.417	0.141	(0.119, 0.164)
	VSM Scoring	**1.281**	(1.14, 1.42)	0.0821	0.099	(0.084, 0.114)
	DNN-LTR Scoring	1.298	(1.15, 1.45)	0.155	**0.219**	(0.188, 0.249)
	Most Frequent	1.351	(1.17, 1.54)	-	0.199	(0.168, 0.229)
	Most Prob. (Person)	1.684	(1.2, 2.17)	0.947	0.207	(0.178, 0.236)
	Most Prob. (Collab)	1.386	(1.17, 1.6)	0.678	0.204	(0.173, 0.234)
	TF.IDF	1.316	(1.15, 1.48)	0.276	0.157	(0.135, 0.178)
Fixed Level (Avg)	Prob. Scoring	1.421	(1.17, 1.68)	0.769	0.139	(0.117, 0.161)
	VSM Scoring	1.333	(1.16, 1.51)	0.385	0.096	(0.081, 0.112)
	DNN-LTR Scoring	**1.281**	(1.14,1.42)	0.082	**0.223**	(0.193, 0.254)
	Most Frequent	1.351	(1.17, 1.54)	-	0.211	(0.182, 0.24)
	Most Prob. (Person)	1.614	(1.19, 2.04)	0.931	0.211	(0.181, 0.241)
	Most Prob. (Collab)	1.333	(1.14, 1.52)	0.401	0.208	(0.178, 0.239)
	TF.IDF	1.298	(1.15, 1.45)	0.155	0.159	(0.138, 0.181)
Plain List-No Grouping	Prob. Scoring	**1.246**	(1.13, 1.36)	0.500	0.157	(0.134, 0.18)
	VSM Scoring	1.263	(1.14, 1.39)	0.832	0.115	(0.097, 0.133)
	LTR Scoring	1.281	(1.14, 1.39)	0.832	0.196	(0.218, 0.278)
	Most Frequent	**1.246**	(1.13, 1.36)	-	0.222	(0.193, 0.251)
	Most Prob. (Person)	1.351	(1.16, 1.54)	0.973	0.227	(0.196, 0.259)
	Most Prob. (Collab)	1.281	(1.13, 1.43)	0.832	**0.229**	(0.196, 0.261)
	TF.IDF	1.263	(1.14, 1.39)	0.832	0.168	(0.143, 0.192)
Plain List with Grouping	Prob. Scoring	1.316	(1.13, 1.5)	0.392	0.156	(0.133, 0.18)
	VSM Scoring	**1.263**	(1.14, 1.39)	0.036*	0.117	(0.098, 0.136)
	DNN-LTR Scoring	1.293	(1.14, 1.46)	0.245	**0.242**	(0.212, 0.271)
	Most Frequent	1.333	(1.16, 1.5)	-	0.216	(0.188, 0.245)
	Most Prob. (Person)	1.491	(1.2, 1.78)	0.936	0.222	(0.192, 0.251)
	Most Prob. (Collab)	1.368	(1.18, 1.56)	0.740	0.226	(0.195, 0.258)
	TF.IDF	1.298	(1.15, 1.45)	0.210	0.169	(0.144, 0.193)

Table 5.6: Comparing impact of different tree construction strategies using #Actions and F-NDCG for the proposed approaches and baselines using TREC-CS dataset (sample size=57), * denotes statistically significant result at p-value<0.05.

Tree Strategy	Scoring Method	#Actions			F-NDCG	
		Mean	95% CI	p-value	Mean	95% CI
Fixed Level (Max)	Prob. Scoring	2.982	(2.83, 3.27)	0.966	0.034	(0.032, 0.036)
	VSM Scoring	2.668	(2.53, 2.92)	0.371	0.038	(0.036, 0.041)
	DNN-LTR Scoring	**2.500**	(2.38, 2.73)	0.051	**0.059**	(0.057, 0.064)
	Most Frequent	2.714	(2.58, 2.97)	-	0.054	(0.052, 0.058)
	Most Prob. (Person)	2.557	(2.43, 2.8)	0.120	**0.059**	(0.057, 0.064)
	Most Prob. (Collab)	2.506	(2.38, 2.75)	0.061	0.053	(0.051, 0.057)
	TF.IDF	3.223	(3.08, 3.51)	0.999	0.035	(0.034, 0.038)
Fixed Level (Avg)	Prob. Scoring	2.823	(2.68, 3.09)	0.933	0.036	(0.035, 0.039)
	VSM Scoring	2.611	(2.48, 2.86)	0.496	0.039	(0.038, 0.042)
	DNN-LTR Scoring	**2.397**	(2.29, 2.61)	0.042*	**0.064**	(0.062, 0.07)
	Most Frequent	2.613	(2.48, 2.86)	-	0.060	(0.057, 0.065)
	Most Prob. (Person)	2.433	(2.32, 2.65)	0.078	0.063	(0.061, 0.068)
	Most Prob. (Collab)	2.520	(2.4, 2.76)	0.240	0.058	(0.056, 0.063)
	TF.IDF	3.205	(3.06, 3.5)	0.999	0.036	(0.034, 0.039)
Plain List-No Grouping	Prob. Scoring	1.841	(1.82, 1.94)	0.991	0.048	(0.047, 0.052)
	VSM Scoring	1.835	(1.81, 1.94)	0.986	0.052	(0.051, 0.056)
	LTR Scoring	1.765	(1.75, 1.85)	0.666	**0.076**	(0.073, 0.081)
	Most Frequent	**1.751**	(1.75, 1.83)	-	0.071	(0.069, 0.077)
	Most Prob. (Person)	1.763	(1.76, 1.85)	0.662	0.074	(0.072, 0.08)
	Most Prob. (Collab)	1.773	(1.77, 1.86)	0.765	0.068	(0.066, 0.073)
	TF.IDF	1.821	(1.8, 1.92)	0.975	0.044	(0.042, 0.047)
Plain List with Grouping	Prob. Scoring	2.147	(2.1, 2.29)	0.069	0.049	(0.047, 0.053)
	VSM Scoring	**1.997**	(1.96, 2.13)	0.0003*	0.052	(0.051, 0.056)
	DNN-LTR Scoring	2.108	(2.08, 2.23)	0.011*	**0.076**	(0.073, 0.082)
	Most Frequent	2.244	(2.2, 2.39)	-	0.071	(0.069, 0.077)
	Most Prob. (Person)	2.117	(2.08, 2.25)	0.019*	0.074	(0.072, 0.08)
	Most Prob. (Collab)	2.155	(2.12, 2.29)	0.073	0.068	(0.066, 0.073)
	TF.IDF	2.388	(2.35, 2.54)	0.985	0.044	(0.042, 0.047)

Table 5.7: Comparing impact of different tree construction strategies using #Actions and F-NDCG for the proposed approaches and baselines using Yelp dataset (sample size=1,495), * denotes statistically significant result at p-value<0.05..

From the table we can see that the baseline Plain List-No Grouping achieved the minimum number of actions and the maximum F-NDCG value. This is expected as the method orders highly relevant t-facets at its top. But the list might contain several t-facets belonging to the same parent which are not grouped together, and which might confuse the reader.

In terms of the number of actions, the next best strategy is the Plain List with grouping, there are two contributing factors to this: the first is that the strategy maintains the top t-facet at top of the tree; the second is that the first page of t-facets contains the same t-facets as in the Plain List-No Grouping. This preserves the most relevant t-facets in the first page and minimize the number of actions while providing a more readable format.

In the case of Fixed Level, both Max and Avg methods performed worse, probably due to the constrain on the fixed number of t-facets included for each level. This allowed some irrelevant t-facets to appear in the first t-facets page and in the same time downgraded other more relevant t-facets. The effect is clearly reflected on the F-NDCG metric as it measures the usefulness of first t-facets page. The Max version of the Fixed Level method performed better than the Avg one as it maintained the top t-facets at the top of the t-facet page.

In order to measure the impact of the tree construction strategy statistically, a one tailed t-test on the overall results of the baselines was performed. In this test H_0 is that using t-facet grouping strategies will not impact the # Actions metric. The results produced for Plain List with No Grouping were used as baseline. The alternative hypothesis H_a is that using Plain List with Grouping strategy impacts the results of the number of action metric, the next alternative hypothesis H_b is that using the Fixed Level (Max) grouping strategy impacts the #Actions metric, and Finally the third hypothesis H_c, is that Fixed Level (Avg) strategy have impact on the results of the number of actions metric. Over the 11 scoring methods (listed in table A1.7 and A1.8) it was found that the three tree construction strategies had a significant impact on the #Actions metric at p-value $< .05$ (n=11, p-value using

Fixed Level (max) for TREC-CS is .004 and for Yelp is .00013; p-value using Fixed Level (avg) for TREC-CS is .004465 and Yelp is .00026; p-value using Plain List with Grouping for TREC-CS is .011 and for Yelp is .000611).

Considering the F-Scan metric (see tables A1.7 and A1.8), the three tree construction methods improved the F-Scan results over the plain list baseline without degrading the F-NDCG values. The reason behind this improvement is how the t-facet pages are organized. For example, if a relevant parent t-facet appeared in the first page, but the desired end-leaf t-facet did not appear in the plain list, the user will scroll down and scan many facets before reaching the required t-facet. But in other t-facet grouping strategies where end-leaf t-facets are grouped by their parent t-facet, the desired end-leaf t-facet will be easily accessible through the 'More' link as soon as the parent t-facet is found, which most probably will appear earlier in the t-facet tree.

5.4 Limitations and Caveats

The three evaluation metrics measure how the ranked t-facets help users better reach to relevant documents. Thus, the assessment of the t-facet ranking is affected by the quality of the underlying document ranking algorithm, and its ability to rank relevant documents higher. Hence, in order to obtain more conclusive results, the the proposed approaches was experimented with using two datasets (TREC-CS, and Yelp) with varying document ranking quality.

As the performance of the search engine deteriorates the t-facet ranking will aid the ranking process by promoting relevant documents. However, existence of too much noise and irrelevant documents will prevent that from happening. This is true especially using the evaluation metrics proposed by INEX 2011 Data Centric task. As the three metrics assume that there is at least one facet that can promote a relevant document in the first page of results.

This case was found in the Yelp dataset where the document ranking engine has

NDCG value equals to 0.1608. In many requests, a relevant search target was not found at the first page of results after filtering with all t-facets. The evaluation framework returned an empty or Null value in this case. To minimize the number of requests returning Null values the number of documents per page was increased from five (like in TREC-CS experimental setup) to ten in Yelp setup. Requests that still return Null after changing the setup were excluded from our evaluation since they reflect the failure of the document ranking engine rather than the t-facet ranking method (they are 34 requests in total).

The choice of #Actions as the main metric somehow mitigated this limitation by focusing only on the number of clicks the users perform on the t-facet list. But it is not entirely mitigated as it still assumes that a relevant result can be found at the first page after filtering with one of the retrieved t-facets. Further investigation is needed to develop a metric similar to the F-Scan, but instead of adding the document position it should exclude it from the equation, because it is the responsibility of the document ranking engine not the facet ranking algorithm.

5.5 Generalizability of the Approach

In this section we discuss the generalizability of the approach from three perspectives: 1) Using the proposed approaches across two datasets within the same domain and across domains. 2) Using the ranking approaches for other types of facets (i.e. p-facets). 3) Using the proposed approaches across different document ranking engines.

First, in this research the proposed approaches were experimented over two datasets, all the parameter tuning was carried out on the TREC-CS dataset. The chosen parameters were then used for the two datasets. This gave consistent results over the two datasets. However, to have more confidence in the methods' performance, further parameter tuning might be needed when moving from one dataset to another. For example, different gamma values for the VSM scoring approach might lead to better results on the Yelp dataset.

Regarding the change of domain, there is no reason to believe that the proposed methods will not suit other precision-oriented search tasks. Given that a number of top baselines were borrowed from other domains. For example, most probable methods were initially developed for the MovieLens dataset in [14], and yet they provided competitive performance on two POI suggestion datasets. In the same way it is presumed that the probabilistic scoring method & VSM & LTR scoring will give reasonable results in the movie recommendation domain. Other domains like e-commerce might involve different considerations regarding the ranking process, this will need further study on the specific domains using these approaches to fully understand their usability.

Turning to the second perspective, the proposed methods are developed to handle the characteristics of special case of the t-facets. It is hard to assume that they will work for property-based facet, most probably a different approach will be required. Two factors are behind this claim, first the p-facets do not necessarily have a hierarchical structure (e.g. book author facet), secondly, even if they have a hierarchical structure (dates for example can be grouped by year, month) the user interests in this case will not be modeled in the same way as for t-facets.

For the third perspective, the proposed methods have the advantage of interoperability across search engines. This is because the probabilistic scoring models and the DNN-LTR methods use the document ranking scores as a black-box. As a result, the proposed methods can be implemented on top of an existing search engine without interfering with how it works.

In addition to that, most of the user-topic profiles, VSM user profiles, other user features can be pre-calculated, indexed offline and updated as needed. This makes these methods scalable to larger datasets. This also makes them easy to compute in run time without generating extra processing above the processing performed by the document search engine.

5.6 Summary of Key Findings

Overall, from the results, we can see that DNN-based models could effectively minimize the user effort on both datasets. Indeed, VSM scoring minimized the # of Actions and F-Scan metrics but performed poorly in terms of F-NDCG, which means that the tree provided in the first page contains more irrelevant facets.

Considering that scoring methods are based on individual user profiles, and that in the TREC-CS dataset, each user profile has either 30 or 60 ratings, it is reasonable to assume that such a small number of ratings impacts negatively on the performance of personalization methods. This number of ratings is not enough to capture the complexity of the user interests.

On the other hand, we can see in the Yelp dataset how feeding the user model with more ratings enhances the model performance. Indeed, Yelp dataset contains more user preference samples and interaction data. This helps in building richer user profiles to aid the personalization process. This is evident not only in our DNN model, but also across all the baseline systems that make use of user profiles, i.e. Prob. Scoring, VSM Scoring and Most Probable (Person). The conclusion here is that as the historical interactions of the users increase the user profile will depict their interests better.

The results have also demonstrated that approaches utilizing the meaning of the t-facet benefited the ranking process. Using topic-based user profiles or the semantic representation of the t-facet was useful during the ranking process. This combined with rich and diverse personal profiles aid the personalized ranking process.

Considering the hierarchical nature of the t-facets, it was found that the way the t-facet tree is organized impacts the user effort needed to find the search target. This research was not concerned with deciding which strategy is the best, it was more concerned with highlighting the need for this step, and that providing the user with a tree structure rather than a ranked list of t-facets needs special handling procedure. Calculating the scores for all the t-facets at all levels adds complexity

to the ranking process, on the other hand, grouping the parent t-facets using the average of the score of their ancestors negatively impacted the evaluation metrics. It is important to carefully choose the appropriate tree construction strategy as part of the ranking approach. The experimental results have demonstrated the effect of the tree construction step on the evaluation metrics is significant.

We also compared the scan cost of our faceted systems to a traditional search engine. The scan cost for a POI search engine without using any facets is 6.15 for TREC-CS and 43.5 for Yelp. It is noteworthy that the reported F-Scan values for all the t-facet search methods (DNN and baselines) outperformed the scan cost incurred by a non-faceted system. This is particularly obvious for the larger Yelp dataset, where the margin of enhancement is larger.

Finally, it was also found that the quality of the facet rank approach depends on the quality of the underlying document ranker. Because of when the t-facet ranking approach happens after the document ranking, it presumes that the document ranking step will be able to retrieve the intended search target. If this search target was not identified nor retrieved by the search engine, no matter how good the facet ranking approach is, it will not be able to help the user in finding this target.

The findings obtained by the evaluation results reported in this chapter answered the research question posed in this thesis in chapter 1, section 1.3. This research investigates if personalizing t-facet ranking, using user historical feedback, can minimize the user effort to reach the intended search target. Using the number of actions as a proxy to user effort, the results have proven that algorithms developed for the type-based facet ranking personalization based on individual user profiles could effectively minimize the user effort needed to reach intended search target.

Chapter 6

Conclusion

This chapter gives a quick recap for the research and experiments covered in this thesis. It starts by revisiting the research question and objectives presented in the first chapter, it discusses to what extent the research question was answered and objectives were met. It also summarizes the findings of the study and lists the main contributions of this work.

This discussion is followed by possible future work and interesting research directions to be pursued based on the outcomes of this work. Finally, the chapter ends with concluding remarks highlighting the applicability of this work from both academic and commercial perspectives.

6.1 Revisiting Research Question, Objectives and Achievements

The research question posed in the first chapter, section 1.3 is:

Can personalizing type-based facet ranking according to user historical feedback minimize the effort needed by users to fulfill their search needs?

In order to answer this research question, this thesis tracked a number of research objectives outlined in section 1.3. Below we list each objective and how it was tackled by this work:

Objective 1

As for the first objective, it was to review the state of the art research related to faceted search systems in general, with a focus on existing facet ranking approaches. To fulfill this objective, a detailed state of the art survey studying existing FSS was conducted and presented in chapter 2. It covered aspects of FSS that affect the facet ranking process. The chapter analysed FSS from users' information need perspective; how they interact with the system and how the underlying data collection is structured. The survey also introduced facets, their types, how they are generated and the different approaches adopted to process them.

Moreover, a detailed review of existing personalized and non-personalized facet ranking methods was conducted. Key published literature related to faceted search was surveyed to find how the facet ranking is performed in section 2.3. In that section, existing ranking methods were described, analysed, and critically evaluated. Finally the discussed methods are summarized in a table for comparison in section 2.5.

Objective 2

The objective was to investigate existing FSS evaluation frameworks, and the most commonly used techniques and metrics to evaluate facet ranking approaches. In order to address the second objective, section 2.4 included the most commonly used evaluation strategies, domains, and datasets to evaluate existing facet ranking approaches. FSS either follows a task-based or simulation-based evaluation. Most of the precision-oriented personalized FSSs follow a simulation-based assessment.

Task-based evaluation can be affected by other aspects of FSS, like the quality of the user interface and responsiveness of the system. Simulation-based evaluation helps focus on the facet ranking step in isolation from other FSS aspects. Furthermore, to

be proven useful, personalization needs to be applied to a large base of users, which is expensive to conduct in a task-based setting and might not yield conclusive results. Understanding how earlier researchers assessed their systems and the challenges they met was instrumental in developing this thesis's experimental setup.

Objective 3

The third research objective was concerned with defining the criteria for dataset appropriateness and its needed structure. One of the outcomes of the literature review was the identification of the lack of available benchmarks for 'personalised' facet ranking tasks. To overcome this and obtain a dataset collection to conduct the experimental evaluation, this thesis suggested a dataset creation framework. This framework utilizes existing collections and customizes it to fit the evaluation of the personalized t-facet ranking task.

The framework, presented in chapter 4, section 4.1, identifies the eligibility criteria for the dataset as well as its designated structure. It also illustrates the dataset customization process in detail and demonstrates how this can be applied in two different scenarios.

Objective 4

As for this objective, it was to choose a domain task and collect suitable data for this task based on the defined criteria. In order to meet this objective, POI suggestion domain was chosen for the experiments included in this thesis. Under this domain two datasets were selected as they met the eligibility criteria outlined by the framework. The first dataset was customized from TREC-CS 2016 collection and presented in section 4.1.3. The second is derived from Yelp Open Dataset and demonstrated in section 4.1.4. By proposing the dataset creation framework and applying it to the two datasets, this work overcomes the research challenge concerned with deciding an appropriate search task (described in Chapter 1, section 1.4.4).

Objective 5

This research objective was to identify, implement and examine existing baseline facet ranking methods and analyse how they perform on the chosen dataset(s) and evaluation framework. One of the outcomes of the state-of-the-art review was the ability to identify key facet ranking baselines. Although, none of them were particularly concerned with type-based facets. However, they could be used as scoring methods for the approach proposed in this thesis. Four non-personalized and four personalized baselines were identified in the literature and included in this thesis's experimental results.

Implementing a variety of baselines provided deeper comprehension of the characteristics of each algorithm. Comparing them with the ranking approaches proposed by this work showed that ranking approaches specifically developed for the type-based facet can outperform generic facet ranking ones. This highlights the need for special algorithms for the specific problem of t-facets ranking.

Objective 6

To tackle the next objective, which was to exploit several personalization strategies from Personalized Information Retrieval (PIR) literature that can be applied to t-facet ranking, this thesis introduced a two-step approach to solve the personalized t-facet ranking problem. The first step of the approach assigns scores to each leaf-node t-facet. Three alternative methods where suggested for this step. Methods are derived from three families of PIR ranking algorithms. The first is based on probabilistic modeling, the second employs vector space modeling, and the third follows a learning to rank technique. The output score reflects the t-facet relevance to the user and input query. This step implements the personalization and leverages the categorical aspects of the t-facets, it addresses the first research challenge of establishing t-facet relevance (described in chapter 1, section 1.4.1).

For the second step, this work explored three alternative strategies to build the final t-facet sub-tree to be provided to the user. The step uses the score generated

for the leaf-node t-facets in order to order their ancestors. This step addresses the hierarchical nature of the t-facets, therefore, it addresses maintaining the multilevel taxonomy structure challenge (described in chapter 1, section 1.4.2). The overall two-step approach, regardless of the scoring method and tree construction strategies, does not require heavy or complex calculations at the t-facet ranking time, the adopted techniques avoid adding complexity to the FSS, which was a challenge that was faced at the beginning of this study (described in chapter 1, section 1.4.3).

Objective 7

In order to meet the next objective, which was to evaluate the proposed ranking approach using well established evaluation methods and metrics, we chose to follow a widely adopted user interaction model for simulation-based evaluation and adapt it to suit our task. Furthermore, this PhD study selected a number of well established metrics to include in the assessment. Metrics were introduced by INEX 2011 Data Centric Track organizers. The complete evaluation setup is detailed in chapter 4, section 4.4. The adopted evaluation strategy tackles the research challenge of evaluation personalized t-facet ranking approaches (described in chapter 1, section 1.4.5).

Objective 8

The last objective of this research was to compare the proposed approach against state of the art baselines and reflect on the most effective system. In order to better understand and assess the behavior of the proposed methods, chapter 5, section 5.3 compares the performance of the proposed methods versus several personalized and non-personalized facet ranking approaches.

The results have demonstrated that our proposed methods outperform existing baselines. This proves that ranking methods developed for the type-based facets minimize the effort needed by the user to reach their search target. It has also shown that the strategy adopted for the final t-facet tree construction affects the ranking metrics. Another key finding was that the long user profiles improves the personalization process. Results analysis and key findings for this comparison in

addition to other experiments of the thesis are summarized in section 5.6.

Fulfilling the eight research objectives helped formulate an answer to the research question posed in this thesis. According to our experiments and well-founded evaluation results, the answer is: Yes, using the proposed approach, which personalizes t-facet rankings based on individual historical user feedback, could effectively minimize the effort needed by the user to reach their intended search target.

Achievements

This thesis provides a coherent, proper and well-established scientific research addressing the personalized type-based facet ranking problem. It acts as a solid foundation on which future research can be conducted. This work resulted in a number of publications in reputable conferences aiming to advance the area of personalized faceted search systems.

- Esraa Ali, Annalina Caputo, Séamus Lawless and Owen Conlan. A Probabilistic Approach to Personalize Type-based Facet Ranking for POI Suggestion. In Proceedings of 21st The International Conference on Web Engineering (ICWE 2021).

- Esraa Ali, Annalina Caputo, Séamus Lawless and Owen Conlan. Personalizing Type-based Facet Ranking using BERT Embeddings. In Proceedings of SEMANTiCS (SEMANTiCS 2021), In Press.

- Esraa Ali, Annalina Caputo, Séamus Lawless and Owen Conlan. Dataset Creation Framework for Personalized Type-based Facet Ranking Tasks Evaluation. In Proceedings of the Twelfth International Conference of the CLEF Association (CLEF 2021), In Press.

- Esraa Ali, Annalina Caputo, Séamus Lawless and Owen Conlan. Where Should I Go? A Deep Learning Approach To Personalize Type-based Facet Ranking for POI Suggestion. In Proceedings of International Conference of Web Information Systems Engineering (WISE 2021).

In addition to those, this work resulted in the construction of two datasets proposed for the evaluation of personalized t-facet ranking tasks. In particular, the Yelp dataset has the potential to be extended to evaluate property-based facet ranking. As the original dataset provides multiple attributes linked to the POIs, therefore they can be used as property-based facets. A publication is underway which documents both datasets in a resource paper, and release the code and all the resources online for other researchers in the field. This publication will target the European Conference on Information Retrieval. The overall outcomes of the thesis are being concluded and summarized in a journal submission to The Journal of Web Semantics, special issue on Knowledge Graphs and Information Retrieval.

6.2 Application Areas and Industrial Impact

The approaches proposed by this work were applied to existing real-life datasets, this proves the approach applicability in several e-tourist commercial websites. Moreover, the approach can inspire t-facet ranking approaches in other industrial domains like e-commerce, where the faceted browsing is the dominant search paradigm.

The proposed scoring methods are also beneficial to other IR tasks in non-faceted search systems. In which case, personalizing the ranking of the types of the documents is desired as part of its results ranking process. Ranking the types or categories of the documents according to prior user interactions is used to improve the ranking of the results. Examples of that are conversational search systems, image search systems and video search systems.

For example, in the SeMantic AnsweR Type prediction (SMART) task [78], participants are asked to predict the type of answer to the question. In this task, types are derived from popular semantic ontologies like Wikidata or DBpedia. This can be further extended to include personalization signals in the prediction procedure.

Query facet expansion is another adjacent research area, in which suggesting the facets to extend the query requires ranking relevant facets to the user. This includes

type-based facets, in some cases, a small tree structure is suggested to help the user during the typing of the query [79]. Our two-step approach can be employed to rank the relevant t-facets and organize them in a tree structure for the user. This is a growing research area which can benefit from the outcomes of this research.

6.3 Future Work

This section puts forward prospective research directions to be pursued based on the outcomes of this thesis.

Improving Personalization

Our experiment has demonstrated that personalization could positively affect the ranking process. It is worthwhile to investigate how more advanced personalization techniques can introduce further improvement. For the time being, this experiment personalizes based upon a static user profile and historical user selections. User behavior resulting from interaction with the system is likely to provide some indication of user intent or current interest and could be included in future experiments.

Furthermore, using advanced user profile representation using knowledge graphs might benefit the t-facet ranking process. This representation should have the capability to encode several aspects of the person in an abstract way. Examples of aspects that can be considered are: user demographics, modeling temporal interests, and networking information, like friend networks or family relationships. This will build richer user profiles and introduce further improvement to the personalization process.

Other Tree construction strategies

The methods provided in this work for the tree construction aim mainly at shedding light on the need for such a step, and how different decisions in grouping and ordering the t-facet ancestors, impact the ranking performance. Other strategies can be adopted for this step in combination with the scoring techniques.

A straightforward approach is using a predefined threshold value to select which t-facets will be displayed in the first page, and then group the selected level-2 t-facets by their parent level-1 t-facet. Level-1 facets are ordered by their top level-2 facet score. The chosen t-facets are displayed in the first page and the rest of the relevant t-facets are made available either by more categories link or more sub-categories links.

Another compelling strategy is to automatically zoom in and out to the most relevant t-facets by skipping t-facets from intermediate levels. The strategy should be able to allocate the most suitable level-N facet nodes and promote it to the main list. The final tree will contain meaningful categories rather than intermediate, less specific, categories. For example, let's consider a three-level taxonomy which contains Restaurants at its top level, Italian Restaurant in its second level, and finally Pizza shop in the third level. In response to a query, a tree containing Restaurants and Pizza shop might be more relevant to an individual than a tree containing Restaurants and Italian Food. On the other hand, if a user explicitly mentioned restaurants in the query, the ranking approach could adjust the final tree to include only sub-taxonomy under the Restaurants node and eradicate irrelevant categories at the same level of Restaurants (like Museums, Gyms...etc).

Single-step approach

An alternative to the current approach is a single step algorithm. The currently proposed approach operates over two steps. Instead, the single step approach would take as an input, the document ranking output and use it to generate the final tree directly. DNN models can be utilized for this purpose: because of their advanced capabilities of merging the two steps, and learn from the target tree structure directly.

A similar approach was followed by Tagliabue et al. [79]: they proposed growing a product category tree for type ahead suggestion. They employed in-session user interaction and query embedding in order to produce the category taxonomy path. This suggested category path is followed by the shoppers to find their intended search target. This method utilizes deep neural networks for both features encoding and the path inference task.

Additional Facet Types

In the future, it is possible to extend the current approach and complement it by developing p-facet ranking methods. As a result, one integrated ranking framework for both types of facets will be generated. This will enable investigating further how employing separate ranking methods for t- and p-facets can enhance the user search experience.

Adding more facet types will require adopting more complex interaction methods for evaluating the approach. The interaction among multiple types of facets requires the selection of multiple facets at once. This and other more advanced scenarios should be considered in both simulated and user-based evaluation setup.

Other Potential Datasets and Domains

While conducting this study, we came across several other datasets that could be suitable for this task. Unfortunately the lack of time and resources prevented further investigation of the eligibility of these datasets. In the venue suggestion domain, one example is Airbnb dataset [1].

It also would be interesting to experiment in other domains, like shopping and e-commerce. One of the existing datasets is Amazon product review dataset [2]. However, it was not possible to include this dataset in the presented experiments as the taxonomy of products was not available online.

It is tempting to study how different personalization approaches behave across domains. As user behavior in shopping websites might be different from e-tourism ones. In the latter, the searcher target might be finding a single venue while the shoppers target might be finding multiple items. This, in addition to other domain characteristics, affects how the ranking happens.

[1] `http://insideairbnb.com/get-the-data.html` , accessed July 2021
[2] `https://jmcauley.ucsd.edu/data/amazon/` , accessed July 2021

Toward Task-based Evaluation

In addition to the aforementioned evaluation strategies, this proposed methodology could also be evaluated using a task-based approach with real users. The initial simulated-based evaluation gave insights about the effectiveness of the approaches. Complementing that with task-based evaluation on a large user base using the best performing methods will highlight their strengths and weaknesses.

Additionally, features could be examined in a more realistic setup. In the current experimental setup the queries are automatically generated from user profiles. Ideally, task-based evaluation will enable realistic queries to be provided by the users. As result, this may improve the document ranking and the query based features in the DNN-LTR approaches. This could bring more benefits as well as pose new challenges for the ranking approach.

6.4 Concluding Remarks

This thesis argues that personalized type-based facet ranking approaches introduce a statistically significant improvement over the best non-personalized facet ranking algorithms. The devised scoring techniques took advantage of the historical user feedback to build a user profile to be used in the personalization process. To generate a final t-facet relevance score, the methods combined this user profile, with additional relevance signals from the dataset structure, the query and the underlying document ranking engine. The results of the proposed methods were not achieved by any traditional personalized facet ranking baseline. This highlights that type-based facet ranking task benefits from algorithms which build upon the categorical aspects of t-facets. Personalized type-based facet ranking was proven to effectively minimize the number of clicks the user had to make, hence the user effort, to reach the intended search targets.

Experiments also demonstrated that following different strategies in building the final t-facet tree is a decision which impacts the evaluation metrics. System designers

should be careful in deciding which strategy to follow according to their use case. Simply choosing to average the score of the leaf-node t-facet in order to rank their ancestors will have a negative impact on the number of actions the user needs to take to reach the sought search result, where choosing to group the top leaf-node t-facets by their parent nodes and make the rest of t-facets accessible through the 'More' link will maintain the most relevant t-facets in the top of the tree, therefore minimizing the number of actions needed by the users to reach their intended search target. The final output of the t-facet ranking method ought to be a readable t-facet tree including the top relevant t-facets. Finally, experiments have also shown the use of facet ranking methods, in general, minimize user effort compared to non-faceted search engines (discussed in chapter 5, section 5.6), since it eliminates the need for users to scroll through seemingly endless results.

Research in this area should pursue more advanced user profiles to include other aspects, like temporal aspects of user dynamics, or in-session interactions of the user. Regarding the t-facets hierarchical nature, while the study of the tree construction is not an objective of this thesis, one of the outcomes of the evaluation was the observation that this step actually has an impact on the adopted metrics.

Finally, this PhD thesis concluded that personalized t-facet ranking methods, based on individual user historical picks, can effectively minimize the effort needed by the users in order to reach their intended search target. This finding answered the posed research question in this thesis. It is hoped by the author of the thesis that this effort advances the research in faceted search systems and provides a coherent reference and a principled starting point to other researchers who wish to explore and extend this research direction.

Bibliography

[1] Qiuyue Wang, Georgina Ramírez, Maarten Marx, Martin Theobald, and Jaap Kamps. Overview of the inex 2011 data-centric track. In Shlomo Geva, Jaap Kamps, and Ralf Schenkel, editors, *Focused Retrieval of Content and Structure*, pages 118–137, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg. ISBN 978-3-642-35734-3.

[2] Chenyan Xiong, Russell Power, and Jamie Callan. Explicit semantic ranking for academic search via knowledge graph embedding. In *Proceedings of International World Wide Web Conference Committee (IW3C2)*, 2017.

[3] Yannis Tzitzikas, Nikos Manolis, and Panagiotis Papadakos. Faceted exploration of rdf/s datasets: a survey. *Journal of Intelligent Information Systems*, pages 1–36, 2017. ISSN 1573-7675. doi: 10.1007/s10844-016-0413-8. URL `http://dx.doi.org/10.1007/s10844-016-0413-8`.

[4] Melike Sah and Vincent Wade. Personalized concept-based search and exploration on the web of data using results categorization. In Philipp Cimiano, Oscar Corcho, Valentina Presutti, Laura Hollink, and Sebastian Rudolph, editors, *The Semantic Web: Semantics and Big Data*, pages 532–547, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg. ISBN 978-3-642-38288-8.

[5] Prajna Upadhyay and Maya Ramanath. Preface++: Faceted retrieval of prerequisites and technical data. In Djoerd Hiemstra, Marie-Francine Moens, Josiane

Mothe, Raffaele Perego, Martin Potthast, and Fabrizio Sebastiani, editors, *Advances in Information Retrieval*, pages 554–558, Cham, 2021. Springer International Publishing. ISBN 978-3-030-72240-1.

[6] Weize Kong and James Allan. Extending faceted search to the general web. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, pages 839–848. ACM, 2014.

[7] Emrah Inan, Paul Thompson, Tim Yates, and Sophia Ananiadou. Hsearch: Semantic search system for workplace accident reports. In Djoerd Hiemstra, Marie-Francine Moens, Josiane Mothe, Raffaele Perego, Martin Potthast, and Fabrizio Sebastiani, editors, *ECIR 2021: Advances in Information Retrieval*, pages 514–519, Cham, 2021. Springer International Publishing. ISBN 978-3-030-72240-1.

[8] Tanmoy Chakraborty, Amrith Krishna, Mayank Singh, Niloy Ganguly, Pawan Goyal, and Animesh Mukherjee. Ferosa: A faceted recommendation system for scientific articles. In *Advances in Knowledge Discovery and Data Mining: 20th Pacific-Asia Conference, PAKDD 2016, Auckland, New Zealand, April 19-22, 2016, Proceedings, Part II*, pages 528–541, 2016.

[9] Biying Kong, Nidhi Rajshree, Alfio Massimiliano Gliozzo, Nicolas Rodolfo Fauceglia, Robert G Farrell, Md Faisal Mahbub Chowdhury, and Anish Mathur. Dynamic faceted search on a document corpus, November 5 2020. US Patent App. 16/399,180.

[10] Eetu Mäkelä, Eero Hyvönen, Samppa Saarela, and Kim Viljanen. Ontoviews – a tool for creating semantic web portals. In Sheila A. McIlraith, Dimitris Plexousakis, and Frank van Harmelen, editors, *The Semantic Web – ISWC 2004: Third International Semantic Web Conference, Hiroshima, Japan, November 7-11, 2004. Proceedings*, pages 797–811, Berlin, Heidelberg, 2004. ISBN 978-3-540-30475-3. doi: 10.1007/978-3-540-30475-3_55. URL `http://dx.doi.org/10.1007/978-3-540-30475-3_55`.

[11] Chengkai Li, Ning Yan, Senjuti B. Roy, Lekhendro Lisham, and Gautam Das. Facetedpedia: Dynamic generation of query-dependent faceted interfaces for wikipedia. In *Proceedings of the 19th International Conference on World Wide Web*, WWW '10, pages 651–660, New York, NY, USA, 2010. ACM. ISBN 978-1-60558-799-8. doi: 10.1145/1772690.1772757. URL `http://doi.acm.org/10.1145/1772690.1772757`.

[12] Eyal Oren, Renaud Delbru, and Stefan Decker. Extending faceted navigation for rdf data. In *International semantic web conference*, pages 559–572. Springer, 2006.

[13] Xi Niu, Xiangyu Fan, and Tao Zhang. Understanding faceted search from data science and human factor perspectives. *ACM Trans. Inf. Syst.*, 37(2), January 2019. ISSN 1046-8188. doi: 10.1145/3284101. URL `https://doi.org/10.1145/3284101`.

[14] Jonathan Koren, Yi Zhang, and Xue Liu. Personalized interactive faceted search. In *Proceedings of the 17th international conference on World Wide Web*, pages 477–486. ACM, 2008.

[15] Michal Tvarozek and Mária Bieliková. Personalized faceted navigation in semantically enriched information spaces,". *Advances in Semantic Media Adaption and Personalization*, 2:181–201, 2009.

[16] Michal Tvarožek and Mária Bieliková. Factic: personalized exploratory search in the semantic web. In *International Conference on Web Engineering*, pages 527–530. Springer, 2010.

[17] Fabian Abel, Ilknur Celik, Geert-Jan Houben, and Patrick Siehndel. Leveraging the semantics of tweets for adaptive faceted search on twitter. *The Semantic Web–ISWC 2011*, pages 1–17, 2011.

[18] Siripinyo Chantamunee, Kok Wai Wong, and Chun Che Fung. An exploration of user–facet interaction in collaborative-based personalized multiple facet selection.

Knowledge-Based Systems, 209:106444, 2020. ISSN 0950-7051. doi: https://doi.org/10.1016/j.knosys.2020.106444. URL http://www.sciencedirect.com/science/article/pii/S0950705120305736.

[19] Esraa Ali, Annalina Caputo, and Séamus Lawless. Entity attribute ranking using learning to rank. In *Proceedings of the First Workshop on Knowledge Graphs and Semantics for Text Retrieval and Analysis (KG4IR 2017)*, pages 19–24. CEUR-WS.org, 2017.

[20] Esraa Ali, Annalina Caputo, and Séamus Lawless. Triple scoring using paragraph vector. In *WSDM Cup 2017: Vandalism Detection and Triple Scoring, CUER Extended Proceedings of WSDM*, 2017.

[21] Seyyed Hadi Hashemi, Charles LA Clarke, Jaap Kamps, Julia Kiseleva, and Ellen M Voorhees. Overview of the trec 2016 contextual suggestion track. In *Proceedings of TREC*, volume 2016, 2016.

[22] Ori Ben-Yitzhak, Nadav Golbandi, Nadav Har'El, Ronny Lempel, Andreas Neumann, Shila Ofek-Koifman, Dafna Sheinwald, Eugene Shekita, Benjamin Sznajder, and Sivan Yogev. Beyond basic faceted search. In *Proceedings of the 2008 International Conference on Web Search and Data Mining*, WSDM '08, page 33–44, New York, NY, USA, 2008. Association for Computing Machinery. ISBN 9781595939272. doi: 10.1145/1341531.1341539. URL https://doi.org/10.1145/1341531.1341539.

[23] Ka-Ping Yee, Kirsten Swearingen, Kevin Li, and Marti Hearst. Faceted metadata for image search and browsing. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 401–408. ACM, 2003.

[24] Susan Dumais. Faceted search. *Encyclopedia of Database Systems*, pages 1103–1109, 2009. doi: 10.1007/978-0-387-39940-9_159. URL https://doi.org/10.1007/978-0-387-39940-9_159.

[25] Rasmus Hahn, Christian Bizer, Christopher Sahnwaldt, Christian Herta, Scott Robinson, Michaela Bürgle, Holger Düwiger, and Ulrich Scheel. Faceted

wikipedia search. In Witold Abramowicz and Robert Tolksdorf, editors, *Business Information Systems: 13th International Conference, BIS 2010, Berlin, Germany, May 3-5, 2010. Proceedings*, pages 1–11, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg. ISBN 978-3-642-12814-1. doi: 10.1007/978-3-642-12814-1_1. URL `http://dx.doi.org/10.1007/978-3-642-12814-1_1`.

[26] Maria-Evangelia Papadaki, N. Spyratos, and Yannis Tzitzikas. Towards interactive analytics over rdf graphs. *Algorithms*, 14:34, 2021.

[27] Andreas Harth. Visinav: A system for visual search and navigation on web data. *Web Semantics: Science, Services and Agents on the World Wide Web*, 8(4): 348–354, 2010.

[28] Max L Wilson, Ryen W White, et al. Evaluating advanced search interfaces using established information-seeking models. *Journal of the American Society for Information Science and Technology*, 60(7):1407–1422, 2009.

[29] Y. Tzitzikas and E. Dimitrakis. Preference-enriched faceted search for voting aid applications. *IEEE Transactions on Emerging Topics in Computing*, PP(99): 1–1, 2016. ISSN 2168-6750. doi: 10.1109/TETC.2016.2633432.

[30] Marcelo Arenas, Bernardo Cuenca Grau, Evgeny Kharlamov, Šarūnas Marciuška, and Dmitriy Zheleznyakov. Faceted search over rdf-based knowledge graphs. *Web Semantics: Science, Services and Agents on the World Wide Web*, 37:55–74, 2016.

[31] T. Le, B. Vo, and T. H. Duong. Personalized facets for semantic search using linked open data with social networks. In *2012 Third International Conference on Innovations in Bio-Inspired Computing and Applications*, pages 312–317, Sep. 2012. doi: 10.1109/IBICA.2012.14.

[32] Evgeny Kharlamov, Luca Giacomelli, Evgeny Sherkhonov, Bernardo Cuenca Grau, Egor V Kostylev, and Ian Horrocks. Semfacet: Making hard faceted search easier. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 2475–2478, 2017.

[33] D. Vandic, S. Aanen, F. Frasincar, and U. Kaymak. Dynamic facet ordering for faceted product search engines. *IEEE Transactions on Knowledge and Data Engineering*, PP(99):1–1, 2017. ISSN 1041-4347. doi: 10.1109/TKDE.2017.2652461.

[34] Damir Vandic, Flavius Frasincar, and Uzay Kaymak. Facet selection algorithms for web product search. In *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management*, pages 2327–2332. ACM, 2013.

[35] John A Bivens, Yu Deng, Kaoutar El Maghraoui, Ruchi Mahindru, HariGovind V Ramasamy, Soumitra Sarkar, and Long Wang. Dynamic faceted search, March 26 2019. US Patent 10,242,103.

[36] Nandana Mihindukulasooriya, Ruchi Mahindru, Md Faisal Mahbub Chowdhury, Yu Deng, Nicolas Rodolfo Fauceglia, Gaetano Rossiello, Sarthak Dash, Alfio Gliozzo, and Shu Tao. Dynamic faceted search for technical support exploiting induced knowledge. In Jeff Z. Pan, Valentina Tamma, Claudia d'Amato, Krzysztof Janowicz, Bo Fu, Axel Polleres, Oshani Seneviratne, and Lalana Kagal, editors, *The Semantic Web – ISWC 2020*, pages 683–699, Cham, 2020. Springer International Publishing. ISBN 978-3-030-62466-8.

[37] Geert-Jan Houben. Facet embeddings for explorative analytics in digital libraries. In *Research and Advanced Technology for Digital Libraries: 21st International Conference on Theory and Practice of Digital Libraries, TPDL 2017, Thessaloniki, Greece, September 18-21, 2017, Proceedings*, volume 10450, page 86. Springer, 2017.

[38] Waleed Ammar, Dirk Groeneveld, Chandra Bhagavatula, Iz Beltagy, Miles Crawford, Doug Downey, Jason Dunkelberger, Ahmed Elgohary, Sergey Feldman, Vu Ha, Rodney Kinney, Sebastian Kohlmeier, Kyle Lo, Tyler Murray, Hsu-Han Ooi, Matthew Peters, Joanna Power, Sam Skjonsberg, Lucy Wang, Chris Wilhelm, Zheng Yuan, Madeleine van Zuylen, and Oren Etzioni. Construction of the literature graph in semantic scholar. In *Proceedings of the 2018 Conference of*

the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 3 (Industry Papers), pages 84–91, New Orleans - Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-3011. URL https://www.aclweb.org/anthology/N18-3011.

[39] Anna Kokolaki and Yannis Tzitzikas. Facetize: An interactive tool for cleaning and transforming datasets for facilitating exploratory search. *ArXiv*, abs/1812.10734, 2018.

[40] Hong Son Nguyen, Hong Phuc Pham, Trong Hai Duong, Thi Phuong Trang Nguyen, and Huynh Minh Triet Le. Personalized facets for faceted search using wikipedia disambiguation and social network. In *Advanced Computational Methods for Knowledge Engineering*, pages 229–241. Springer, 2016.

[41] Mohammed Najah Mahdi, Abdul Rahim Ahmad, and Roslan Ismail. Improving faceted search results for web-based information exploration. *International Journal on Advanced Science, Engineering and Information Technology*, 10(3): 1143–1152, 2020.

[42] Siripinyo Chantamunee, Kok Wai Wong, and Chun Che Fung. Collaborative filtering for personalised facet selection. In *Proceedings of the 10th International Conference on Advances in Information Technology*, IAIT 2018, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450365680. doi: 10.1145/3291280.3291796. URL https://doi.org/10.1145/3291280.3291796.

[43] Siripinyo Chantamunee, Kok Wai Wong, and Chun Che Fung. Deep autoencoder on personalized facet selection. In Tom Gedeon, Kok Wai Wong, and Minho Lee, editors, *Neural Information Processing*, pages 314–322, Cham, 2019. Springer International Publishing. ISBN 978-3-030-36808-1.

[44] Marti A. Hearst. Clustering versus faceted categories for information exploration. *Commun. ACM*, 49(4):59–61, April 2006. ISSN 0001-0782. doi: 10.1145/1121949.1121983. URL https://doi.org/10.1145/1121949.1121983.

[45] K Latha, K Rathna Veni, and R Rajaram. Afgf: An automatic facet generation framework for document retrieval. In *International Conference on Advances in Computer Engineering (ACE), 2010*, pages 110–114. IEEE, 2010.

[46] Raffael Affolter and Andreas Weiler. Facetx: Dynamic facet generation for advanced information filtering of search results. In *EDBT/ICDT Workshops*, 2020.

[47] Wisam Dakka, Panagiotis G. Ipeirotis, and Kenneth R. Wood. Automatic construction of multifaceted browsing interfaces. In *Proceedings of the 14th ACM International Conference on Information and Knowledge Management*, CIKM '05, page 768–775, New York, NY, USA, 2005. Association for Computing Machinery. ISBN 1595931406. doi: 10.1145/1099554.1099738. URL https://doi.org/10.1145/1099554.1099738.

[48] José Luis Sánchez-Cervantes, Giner Alor-Hernández, Mario Andrés Paredes-Valverde, Lisbeth Rodríguez-Mazahua, and Rafael Valencia-García. Nala-search: A multimodal, interaction-based architecture for faceted search on linked open data. *Journal of Information Science*, 0(0):0165551520930918, 2020. doi: 10.1177/0165551520930918. URL https://doi.org/10.1177/0165551520930918.

[49] Pavlos Fafalios, Ioannis Kitsos, Yannis Marketakis, Claudio Baldassarre, Michail Salampasis, and Yannis Tzitzikas. Web searching with entity mining at query time. In *Information Retrieval Facility Conference*, pages 73–88. Springer, 2012.

[50] Ioannis Kitsos, Kostas Magoutis, and Yannis Tzitzikas. Scalable entity-based summarization of web search results using mapreduce. *Distributed and Parallel Databases*, 32(3):405–446, 2014.

[51] m.c. schraefel, Max Wilson, Alistair Russell, and Daniel A. Smith. Mspace: Improving information access to multimedia domains with multimodal exploratory search. *Commun. ACM*, 49(4):47–49, April 2006. ISSN 0001-0782. doi: 10.1145/1121949.1121980. URL https://doi.org/10.1145/1121949.1121980.

[52] Evgeny Kharlamov, Luca Giacomelli, Evgeny Sherkhonov, Bernardo Cuenca

Grau, Egor V Kostylev, and Ian Horrocks. Ranking, aggregation, and reachability in faceted search with semfacet. In *Proceedings of the ISWC 2017 Posters & Demonstrations and Industry Tracks co-located with 16th International Semantic Web Conference (ISWC 2017), Vienna, Austria, October 23rd - to - 25th, 2017*, volume 1963 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2017. URL `http://ceur-ws.org/Vol-1963/paper650.pdf`.

[53] Abhijith Kashyap, Vagelis Hristidis, and Michalis Petropoulos. Facetor: Cost-driven exploration of faceted query results. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, CIKM '10, page 719–728, New York, NY, USA, 2010. Association for Computing Machinery. ISBN 9781450300995. doi: 10.1145/1871437.1871530. URL `https://doi.org/10.1145/1871437.1871530`.

[54] Leila Feddoul, Sirko Schindler, and Frank Löffler. Automatic facet generation and selection over knowledge graphs. In Maribel Acosta, Philippe Cudré-Mauroux, Maria Maleshkova, Tassilo Pellegrini, Harald Sack, and York Sure-Vetter, editors, *Semantic Systems. The Power of AI and Knowledge Graphs*, pages 310–325, Cham, 2019. Springer International Publishing. ISBN 978-3-030-33220-4.

[55] Michael Glass, Md Faisal Mahbub Chowdhury, Yu Deng, Ruchi Mahindru, Nicolas Rodolfo Fauceglia, Alfio Gliozzo, and Nandana Mihindukulasooriya. Dynamic facet selection by maximizing graded relevance. *InterNLP 2021*, page 32, 2021.

[56] Sonya Liberman and Ronny Lempel. Approximately optimal facet value selection. *Science of Computer Programming*, 94:18 – 31, 2014. ISSN 0167-6423. doi: https://doi.org/10.1016/j.scico.2013.07.019. URL `http://www.sciencedirect.com/science/article/pii/S0167642313001937`. Web Technologies: Selected and extended papers from WT ACM SAC 2012.

[57] Prajna Upadhyay and Maya Ramanath. Preface: Faceted retrieval of prerequisites using domain-specific knowledge bases. In Jeff Z. Pan, Valentina Tamma, Claudia d'Amato, Krzysztof Janowicz, Bo Fu, Axel Polleres, Oshani Seneviratne,

and Lalana Kagal, editors, *The Semantic Web – ISWC 2020*, pages 601–618, Cham, 2020. Springer International Publishing. ISBN 978-3-030-62419-4.

[58] Roelof Van Zwol, Börkur Sigurbjornsson, Ramu Adapala, Lluis Garcia Pueyo, Abhinav Katiyar, Kaushal Kurapati, Mridul Muralidharan, Sudar Muthu, Vanessa Murdock, Polly Ng, et al. Faceted exploration of image search results. In *Proceedings of the 19th international conference on World wide web*, pages 961–970. ACM, 2010.

[59] Roelof van Zwol, Lluís Garcia Pueyo, Mridul Muralidharan, and Börkur Sigurbjörnsson. Machine learned ranking of entity facets. In *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '10, page 879–880, New York, NY, USA, 2010. Association for Computing Machinery. ISBN 9781450301534. doi: 10.1145/1835449.1835662. URL `https://doi-org.elib.tcd.ie/10.1145/1835449.1835662`.

[60] Roelof van Zwol, Lluis Garcia Pueyo, Mridul Muralidharan, and Borkur Sigurbjornsson. Ranking entity facets based on user click feedback. In *Semantic Computing (ICSC), 2010 IEEE Fourth International Conference on*, pages 192–199. IEEE, 2010.

[61] Chen He, Luana Micallef, Barış Serim, Tung Vuong, Tuukka Ruotsalo, and Giulio Jacucci. Interactive visual facets to support fluid exploratory search. *arXiv preprint arXiv:2108.00920*, 2021.

[62] Róbert Móro, Mária Bieliková, and Roman Burger. *Facet Tree for Personalized Web Documents Organization*, pages 372–387. Springer International Publishing, Cham, 2014. ISBN 978-3-319-11749-2. doi: 10.1007/978-3-319-11749-2_28. URL `https://doi.org/10.1007/978-3-319-11749-2_28`.

[63] Taro Aso, Toshiyuki Amagasa, and Hiroyuki Kitagawa. A system for relation-oriented faceted search over knowledge bases. *International Journal of Web Information Systems*, 2021.

[64] Mostafa Bayomi and Séamus Lawless. Adapt_tcd: An ontology-based context aware approach for contextual suggestion. In *TREC*, 2016.

[65] Mohammad Aliannejadi, Ida Mele, and Fabio Crestani. A cross-platform collection for contextual suggestion. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '17, pages 1269–1272, New York, NY, USA, 2017. ACM. ISBN 978-1-4503-5022-8. doi: 10.1145/3077136.3080752. URL `http://doi.acm.org/10.1145/3077136.3080752`.

[66] Suraj Agrawal, Dwaipayan Roy, and Mandar Mitra. Tag embedding based personalized point of interest recommendation system. *arXiv preprint arXiv:2004.06389*, 2020.

[67] Anne Schuth and Maarten Marx. Evaluation methods for rankings of facetvalues for faceted search. In Pamela Forner, Julio Gonzalo, Jaana Kekäläinen, Mounia Lalmas, and Marteen de Rijke, editors, *Multilingual and Multimodal Information Access Evaluation*, pages 131–136, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg. ISBN 978-3-642-23708-9.

[68] David Sontag, Kevyn Collins-Thompson, Paul N. Bennett, Ryen W. White, Susan Dumais, and Bodo Billerbeck. Probabilistic models for personalizing web search. In *Proceedings of the 5th ACM International Conference on Web Search and Data Mining*, WSDM '12, page 433–442, NY, USA, 2012. ACM. doi: 10.1145/2124295.2124348.

[69] Pierpaolo Basile, Annalina Caputo, and Giovanni Semeraro. Negation for document re-ranking in ad-hoc retrieval. In *Conference on the Theory of Information Retrieval*, pages 285–296. Springer, 2011.

[70] Junmei Wang, Min Pan, Tingting He, Xiang Huang, Xueyan Wang, and Xinhui Tu. A pseudo-relevance feedback framework combining relevance matching and semantic matching for information retrieval. *Information Processing & Management*, 57(6):102342, 2020.

[71] Min Pan, Jimmy Xiangji Huang, Tingting He, Zhiming Mao, Zhiwei Ying, and Xinhui Tu. A simple kernel co-occurrence-based enhancement for pseudo-relevance feedback. *Journal of the Association for Information Science and Technology*, 71(3):264–281, 2020.

[72] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2018.

[73] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *ICLR 2015*, 9, 2015.

[74] Adriel Dean-Hall, Charles L. A. Clarke, Jaap Kamps, Julia Kiseleva, and Ellen M. Voorhees. Overview of the TREC 2015 contextual suggestion track. In Ellen M. Voorhees and Angela Ellis, editors, *Proceedings of The Twenty-Fourth Text REtrieval Conference, TREC 2015, Gaithersburg, Maryland, USA, November 17-20, 2015*, volume 500-319 of *NIST Special Publication*. National Institute of Standards and Technology (NIST), 2015. URL `http://trec.nist.gov/pubs/trec24/papers/Overview-CX.pdf`.

[75] Mohammad Aliannejadi and Fabio Crestani. Personalized context-aware point of interest recommendation. *ACM Trans. Inf. Syst.*, 36(4), October 2018. ISSN 1046-8188. doi: 10.1145/3231933. URL `https://doi.org/10.1145/3231933`.

[76] Anirban Chakraborty, Debasis Ganguly, Annalina Caputo, and Séamus Lawless. A factored relevance model for contextual point-of-interest recommendation. In *Proceedings of the 2019 ACM SIGIR International Conference on Theory of Information Retrieval*, ICTIR '19, page 157–164, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450368810. doi: 10.1145/3341981.3344230. URL `https://doi.org/10.1145/3341981.3344230`.

[77] Nicolas Hug. Surprise: A python library for recommender systems. *Journal of Open Source Software*, 5(52):2174, 2020. doi: 10.21105/joss.02174. URL `https://doi.org/10.21105/joss.02174`.

[78] Nandana Mihindukulasooriya, Mohnish Dubey, Alfio Gliozzo, Jens Lehmann, Axel-Cyrille Ngonga Ngomo, and Ricardo Usbeck. Semantic answer type prediction task (smart) at iswc 2020 semantic web challenge. *arXiv preprint arXiv:2012.00555*, 2020.

[79] Jacopo Tagliabue, Bingqing Yu, and Marie Beaulieu. How to grow a (product) tree: Personalized category suggestions for eCommerce type-ahead. In *Proceedings of The 3rd Workshop on e-Commerce and NLP*, pages 7–18, Seattle, WA, USA, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.ecnlp-1.2. URL `https://www.aclweb.org/anthology/2020.ecnlp-1.2`.

Appendix A1

Reporting Detailed Results

In this appendix we include the complete results obtained for the implemented scoring methods and baselines. Results are reported using the four tree construction strategies introduced in chapter 3, section 3.4. For the sake of completeness, and better comparability, the earlier reported results for all the metrics and methods are included again below. In the tables we highlight the best result values in bold, this is done for each tree construction method.

A1.1 Approach Results

A1.1.1 Probabilistic Scoring Methods

Table A1.1: Probabilistic scoring models results for TREC-CS 2016.

Tree Method	Scoring Method	TREC-CS		
		F-Scan	#Actions	F-NDCG
Fixed Levels (Max)	Model 1 - Exact	4.316	1.544	**0.169**
	Model 1 - Cosine	**3.456**	**1.333**	0.141
	Model 2 - Exact	4.596	1.719	0.119
	Model 2 - Cosine	4.895	1.754	0.107
Fixed Levels (Avg)	Model 1 - Exact	4.105	1.526	**0.178**
	Model 1 - Cosine	**3.561**	**1.421**	0.139
	Model 2 - Exact	4.386	1.719	0.125
	Model 2 - Cosine	4.93	1.807	0.098
Plain List - No Grouping	Model 1 - Exact	7.491	1.281	**0.201**
	Model 1 - Cosine	**3.632**	**1.246**	0.157
	Model 2 - Exact	8.035	1.281	0.144
	Model 2 - Cosine	7.035	1.316	0.115
Plain List with Grouping	Model 1 - Exact	4.211	1.474	**0.198**
	Model 1 - Cosine	**3.614**	**1.316**	0.156
	Model 2 - Exact	4.561	1.509	0.145
	Model 2 - Cosine	4.491	1.579	0.116

Table A1.2: Probabilistic scoring models results for Yelp.

Tree Method	Scoring Method	Yelp		
		F-Scan	#Actions	F-NDCG
Fixed Levels (Max)	Model 1 - Exact	10.605	**2.913**	**0.035**
	Model 1 - Cosine	**10.326**	2.982	0.034
	Model 2 - Exact	13.104	3.443	0.014
	Model 2 - Cosine	12.942	3.613	0.024
Fixed Levels (Avg)	Model 1 - Exact	10.151	**2.781**	**0.038**
	Model 1 - Cosine	**9.701**	2.823	0.036
	Model 2 - Exact	12.668	3.365	0.015
	Model 2 - Cosine	12.726	3.542	0.024
Plain List-No Grouping	Model 1 - Exact	**16.564**	**1.794**	0.042
	Model 1 - Cosine	18.199	1.841	**0.048**
	Model 2 - Exact	25.146	1.833	0.016
	Model 2 - Cosine	29.161	1.918	0.028
Plain List with Grouping	Model 1 - Exact	**10.254**	2.252	0.043
	Model 1 - Cosine	10.440	**2.147**	**0.049**
	Model 2 - Exact	12.168	2.609	0.029
	Model 2 - Cosine	11.884	2.589	0.016

A1.1.2 VSM Scoring Methods

Table A1.3: Results for VSM scoring models for TREC-CS 2016.

Tree Method	Scoring Method	TREC-CS		
		F-Scan	#Actions	F-NDCG
Fixed Levels (Max)	VSM-Rocchio	**3.281**	**1.281**	0.099
	VSM-Ortho	3.754	1.421	**0.133**
	VSM-KNN	4.825	1.667	0.067
Fixed Levels (Avg)	VSM-Rocchio	**3.351**	**1.333**	0.096
	VSM-Ortho	3.912	1.386	**0.140**
	VSM-KNN	4.825	1.667	0.067
Plain List-No Grouping	VSM-Rocchio	**3.526**	**1.263**	0.115
	VSM-Ortho	3.807	1.298	**0.150**
	VSM-KNN	9.632	1.316	0.055
Plain List with Grouping	VSM-Rocchio	**3.281**	**1.263**	0.117
	VSM-Ortho	3.719	1.333	**0.149**
	VSM-KNN	4.877	1.561	0.055

Table A1.4: Results for VSM scoring models for Yelp.

Tree Method	Scoring Method	Yelp		
		F-Scan	#Actions	F-NDCG
Fixed Levels (Max)	VSM-Rocchio	**9.331**	**2.668**	**0.038**
	VSM-Ortho	10.872	3.011	**0.038**
	VSM-KNN	15.611	4.195	0.012
Fixed Levels (Avg)	VSM-Rocchio	**9.225**	**2.611**	**0.039**
	VSM-Ortho	10.688	3.001	0.035
	VSM-KNN	15.611	4.195	0.012
Plain List - No Grouping	VSM-Rocchio	**15.571**	**1.835**	**0.052**
	VSM-Ortho	21.704	1.889	0.043
	VSM-KNN	25.109	1.872	0.022
Plain List with Grouping	VSM-Rocchio	**9.861**	**1.997**	**0.052**
	VSM-Ortho	11.072	2.264	0.042
	VSM-KNN	15.561	2.862	0.022

A1.1.3 DNN-LTR Scoring Methods

Table A1.5: Results for LTR scoring models for TREC-CS 2016.

Tree Method	Scoring Method	TREC-CS		
		F-Scan	#Actions	F-NDCG
Fixed Levels (Max)	Group-All	3.526	1.368	**0.223**
	Group-P	3.474	1.333	0.222
	Group-CF	**3.105**	1.316	0.210
	Group-Q	3.175	**1.298**	0.219
	Group-P+CF	3.298	1.316	0.221
	Group-Q+CF	**3.105**	1.333	0.214
	Group-P+Q	4.263	1.579	0.179
Fixed Levels (Avg)	Group-All	3.404	1.316	0.227
	Group-P	3.526	1.333	**0.230**
	Group-CF	3.123	1.316	0.209
	Group-Q	3.175	1.281	0.223
	Group-P+CF	3.123	**1.263**	0.223
	Group-Q+CF	**3.070**	1.281	0.215
	Group-P+Q	3.772	1.544	0.181
Plain List - No Grouping	Group-All	3.632	**1.246**	0.247
	Group-P	4.702	**1.246**	**0.252**
	Group-CF	**3.298**	1.263	0.233
	Group-Q	3.895	1.263	0.248
	Group-P+CF	3.509	**1.246**	0.249
	Group-Q+CF	3.702	**1.246**	0.232
	Group-P+Q	5.088	**1.246**	0.209
Plain List with Grouping	Group-All	3.404	1.316	0.241
	Group-P	3.614	1.316	**0.242**
	Group-CF	3.105	**1.281**	0.228
	Group-Q	3.175	1.298	**0.242**
	Group-P+CF	**3.088**	1.333	0.239
	Group-Q+CF	3.158	**1.281**	0.227
	Group-P+Q	4.053	1.439	0.199

Table A1.6: Results for LTR scoring models for Yelp Dataset.

Tree Method	Scoring Method	Yelp		
		F-Scan	#Actions	F-NDCG
Fixed Levels (Max)	Group-All	9.964	**2.480**	0.058
	Group-P	**9.903**	2.500	**0.059**
	Group-CF	10.401	2.530	0.054
	Group-Q	10.353	2.505	0.056
	Group-P+CF	9.968	2.503	**0.059**
	Group-Q+CF	10.455	2.536	0.054
	Group-P+Q	9.993	2.512	**0.059**
Fixed Levels (Avg)	Group-All	9.288	2.427	0.064
	Group-P	**9.189**	**2.397**	0.064
	Group-CF	9.608	2.500	0.059
	Group-Q	9.613	2.484	0.061
	Group-P+CF	9.254	2.425	0.064
	Group-Q+CF	9.708	2.518	0.059
	Group-P+Q	9.219	2.426	**0.065**
Plain List - No Grouping	Group-All	12.466	**1.748**	0.075
	Group-P	**12.443**	1.765	**0.076**
	Group-CF	13.205	1.767	0.070
	Group-Q	13.217	1.768	0.070
	Group-P+CF	12.446	1.763	0.075
	Group-Q+CF	13.378	1.781	0.069
	Group-P+Q	12.452	1.761	0.075
Plain List with Grouping	Group-All	**9.775**	2.127	0.075
	Group-P	9.779	**2.108**	**0.076**
	Group-CF	10.098	2.164	0.070
	Group-Q	10.084	2.158	0.070
	Group-P+CF	9.797	2.129	0.075
	Group-Q+CF	10.231	2.175	0.069
	Group-P+Q	9.845	2.134	0.075

A1.2 Comparing Approach Results with Baselines

Table A1.7: Comparing proposed approach and baselines for TREC-CS 2016.

Tree Strategy	Scoring Method	TREC-CS		
		F-Scan	#Actions	F-NDCG
Fixed Level (Max)	Prob. Scoring	3.456	1.333	0.141
	VSM Scoring	3.281	**1.281**	0.099
	DNN-LTR Scoring	**3.175**	1.298	**0.219**
	Most Frequent	3.263	1.351	0.199
	Set-Cover	3.491	1.386	0.187
	Alphabetical	3.614	1.649	0.107
	SemFacet	5.193	1.824	0.021
	MF-SVM	3.947	1.474	0.101
	Most Prob. (Person)	4.053	1.684	0.207
	Most Prob. (Collab)	3.368	1.386	0.204
	TF.IDF	3.298	1.316	0.157
Fixed Level (Avg)	Prob. Scoring	3.561	1.421	0.139
	VSM Scoring	3.351	1.333	0.096
	DNN-LTR Scoring	**3.175**	**1.281**	**0.223**
	Most Frequent	3.175	1.351	0.211
	Set-Cover	3.316	1.351	0.189
	Alphabetical	3.614	1.649	0.107
	SemFacet	5.298	1.860	0.018
	MF-SVM	3.912	1.491	0.11
	Most Prob. (Person)	3.737	1.614	0.211
	Most Prob. (Collab)	3.351	1.333	0.208
	TF.IDF	3.298	1.298	0.159
Plain List - No Grouping	Prob. Scoring	3.632	**1.246**	0.157
	VSM Scoring	**3.526**	1.263	0.115
	DNN-LTR Scoring	3.895	1.263	**0.248**
	Most Frequent	3.386	1.246	0.222
	Set-Cover	3.474	1.281	0.208
	Alphabetical	5.561	1.281	0.089
	SemFacet	14.877	1.351	0.013
	MF-SVM	4.263	1.351	0.121
	Most Prob. (Person)	4.895	1.351	0.227
	Most Prob. (Collab)	3.649	1.281	0.229
	TF.IDF	4.474	1.263	0.168
Plain List with Grouping	Prob. Scoring	3.614	1.316	0.156
	VSM Scoring	3.281	**1.263**	0.117
	DNN-LTR Scoring	3.175	1.298	**0.242**
	Most Frequent	3.228	1.333	0.216
	Set-Cover	3.368	1.368	0.201
	Alphabetical	4.298	1.438	0.089
	SemFacet	5.228	1.631	0.013
	MF-SVM	3.877	1.316	0.121
	Most Prob. (Person)	3.93	1.491	0.222
	Most Prob. (Collab)	**3.105**	1.368	0.226
	TF.IDF	3.386	1.298	0.169

Table A1.8: Comparing proposed approach and baselines for Yelp.

Tree Strategy	Scoring Method	Yelp		
		F-Scan	#Actions	F-NDCG
Fixed Level (Max)	Prob. Scoring	10.326	2.982	0.034
	VSM Scoring	**9.331**	2.668	0.038
	DNN-LTR Scoring	9.903	**2.500**	**0.059**
	Most Frequent	10.934	2.714	0.054
	Set-Cover	10.763	2.868	0.049
	Alphabetical	21.676	5.893	0.007
	SemFacet	19.085	4.822	0.006
	MF-SVM	19.29	4.729	0.007
	Most Prob. (Person)	9.655	2.557	**0.059**
	Most Prob. (Collab)	10.498	2.506	0.053
	TF.IDF	11.605	3.223	0.035
Fixed Level (Avg)	Prob. Scoring	9.701	2.823	0.036
	VSM Scoring	9.225	2.611	0.039
	DNN-LTR Scoring	9.189	**2.397**	**0.064**
	Most Frequent	9.985	2.613	0.060
	Set-Cover	10.065	2.765	0.053
	Alphabetical	21.676	5.893	0.007
	SemFacet	18.039	4.800	0.005
	MF-SVM	18.165	4.659	0.008
	Most Prob. (Person)	**9.097**	2.433	0.063
	Most Prob. (Collab)	9.738	2.520	0.058
	TF.IDF	11.452	3.205	0.036
Plain List - No Grouping	Prob. Scoring	18.199	1.841	0.048
	VSM Scoring	15.571	1.835	0.052
	DNN-LTR Scoring	12.443	1.765	**0.076**
	Most Frequent	13.417	1.751	0.071
	Set-Cover	16.795	1.771	0.067
	Alphabetical	52.013	2.004	0.006
	SemFacet	53.098	1.926	0.004
	MF-SVM	46.302	1.965	0.006
	Most Prob. (Person)	**12.401**	**1.763**	0.074
	Most Prob. (Collab)	12.868	1.773	0.068
	TF.IDF	18.516	1.821	0.044
Plain List with Grouping	Prob. Scoring	10.440	2.147	0.049
	VSM Scoring	9.861	**1.997**	0.052
	DNN-LTR Scoring	9.779	2.108	**0.076**
	Most Frequent	10.638	2.244	0.071
	Set-Cover	10.179	2.312	0.067
	Alphabetical	28.633	3.642	0.006
	SemFacet	17.095	3.319	0.004
	MF-SVM	16.36	3.342	0.006
	Most Prob. (Person)	**9.496**	2.117	0.074
	Most Prob. (Collab)	10.121	2.155	0.068
	TF.IDF	11.728	2.388	0.044

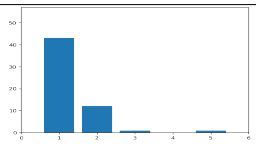
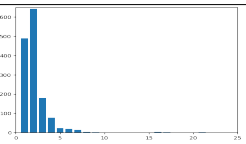
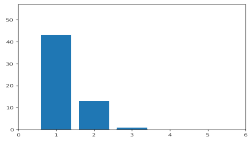
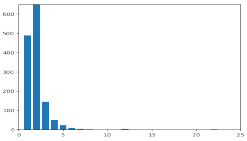
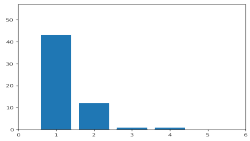
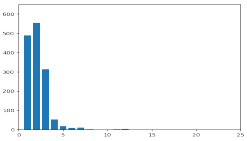
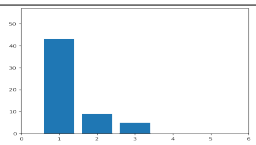
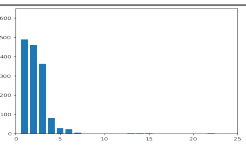
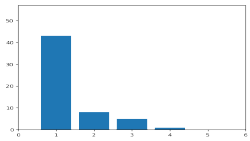
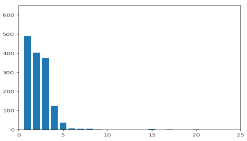
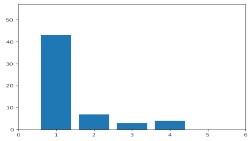
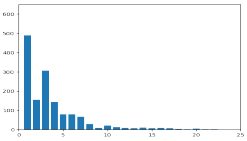
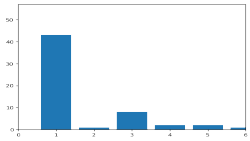
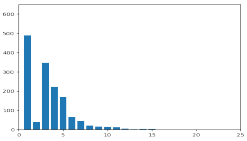
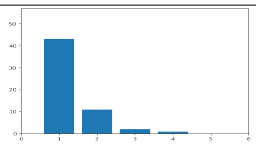
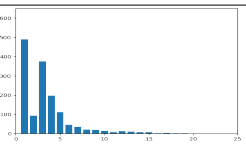
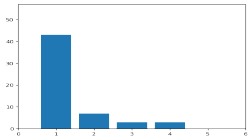
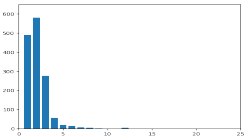
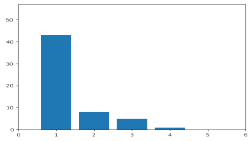
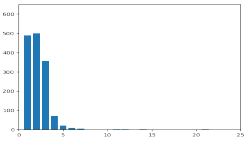
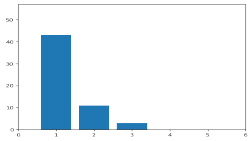
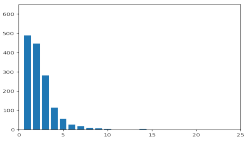
Scoring Method	Scoring Method	TREC-CS	Yelp
Our personalized methods	Prob. Scoring		
	VSM Scoring		
	DNN-LTR Scoring		
Non-personalized	Most Frequent		
	Set-Cover		
	Alphabetical		
	SemFacet		
Personalized	MF-SVM		
	Most Prob. (Person)		
	Most Prob. (Collab)		
	TF.IDF		

Table A1.9: Histograms for #Actions for the proposed approaches and baselines using Plain List with Grouping strategy.