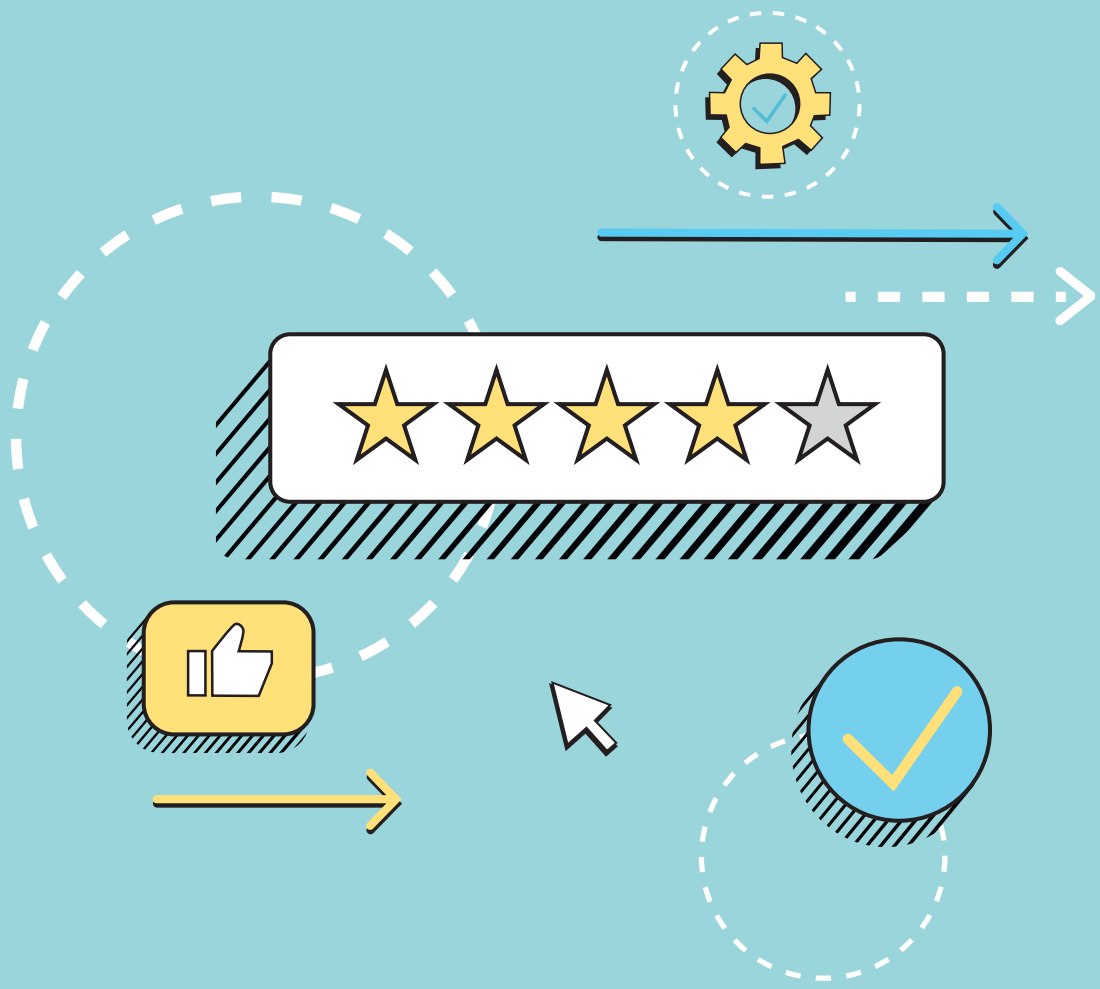
Spotlight on Recommender Systems:

Contributions to Selected Components in the Recommendation Pipeline



Institute for Computing and Information Sciences

Gebrekirstos Gebreselassie Gebremeskel

RADBOUD UNIVERSITY PRESS

Radboud Dissertation Series

Spotlight on Recommender Systems:

Contributions to Selected Components in the Recommendation Pipeline

The research reported in this thesis has been mostly carried out at CWI, the Dutch National Research Laboratory for Mathematics and Computer Science, within the information Access Group.

SIKS Dissertation Series No. 2025-31. The research reported in this thesis has been carried out under the auspices of SIKS, the Dutch Research School for Information and Knowledge Systems.

Part of this work was supported by COMMIT/ A public-private research community.

Spotlight on Recommender Systems: Contributions to Selected Components in the Recommendation Pipeline
Gebrekirstos Gebreselassie Gebremeskel

Radboud Dissertation Series
ISSN: 2950-2772 (Online); 2950-2780 (Print)

Published by RADBOUD UNIVERSITY PRESS
Postbus 9100, 6500 HA Nijmegen, The Netherlands
www.radbouduniversitypress.nl

Design: Gebrekirstos Gebreselassie Gebremeskel
Cover: Proefschrift AIO | Guntra Laivacuma
Printing: DPN Rikken/Pumbo

ISBN: 9789465151137
DOI: 10.54195/9789465151137
Free download at: https://doi.org/10.54195/9789465151137

Spotlight on Recommender Systems:

Contributions to Selected Components in the Recommendation Pipeline

Proefschrift

ter verkrijging van de graad van doctor
aan de Radboud University Nijmegen
op gezag van de rector magnificus prof. dr. J.M. Sanders,
volgens besluit van het college voor promoties
in het openbaar te verdedigen op

dinsdag 8 juli 2025
om 16.30 uur precies

door

Gebrekirstos Gebreselassie Gebremeskel

geboren op 5 augustus 1984
te Hawzien (Ethiopië)

Promotor: Prof. dr. ir. A.P. de Vries

Manuscriptcommissie:

Prof. dr. ir. D. Hiemstra	
Prof. dr. M.A. Larson	
Prof. dr. A.P.J. van den Bosch	(Universiteit Utrecht)
Prof. dr. J. van Ossenbruggen	(Vrije Universiteit Amsterdam)
Prof. dr. R. Baeza-Yates	(Northeastern University, US)

Dedicated to my beloved nephew Gezu Mezgebe and my cousin Weldegebriel Tekhlai—two young souls who gave their lives resisting the Tigray Genocide. To my grandmother, Abrehet Gidey, who passed away during the cruel blockade on Tigray. To the millions of victims of the Tigray Genocide. And to all who bravely fought and stood against this atrocity.

Contents

List of Figures

List of Tables

1

Introduction

The advent of the Internet resulted in the unprecedented explosion of information and the subsequent need for new methods and tools to access information. Information Retrieval (IR), as a field of study, expanded to address this need. Flagship applications of this research area are search engines such as Google, Bing or Baidu. IR is a broad research area consisting of several processes that culminate in the provision of an answer (or answers) to an information need. For a long time, IR was dominated by the sense of pull triggered by the user issuing a query. IR operates from the assumption that the user knows what he/she wants and/or what query to issue.

The sense of pull (initiated by the issuance of a query) that characterizes IR was soon joined by a sense of push where Information Access (IA) systems proactively supply information to the user without the user entering queries. These push systems are broadly called Recommender Systems (RS). The emergence of push systems is motivated by 1) an understanding that it is not easy for users to translate their information needs to queries, and 2) the interest to influence users or advertise products and services. In reality, IR and RS systems are mixed and used together in daily information provision and consumption.

Recommendation is ubiquitous in today's digital era. From e-commerce sites to search engines and to news portals, recommendation is an integral part of the information provision and consumption, and revenue. A recommender system has three conspicuous components: a target user profile, a collection, and a recommendation engine. The user profile is the entity for which recommendation is to be provided. This can vary from a Wikipedia entity profile for which relevant information is recommended for update, to a prototypical user (of an e-commerce website or a news portal) for whom items are provided.

The collection is the information haystack, so to speak, from which recommendation items are picked out. This can be a stream of news items, a collection of product items, or a list of websites. The size of the collection vis-à-vis the reality that only a small part of it is of relevance or interest to the target user profile in question is the main raison d'être for recommender systems. In the absence of RS systems, the user has to contend with the overwhelming collection.

The third component is the recommendation engine, which takes the user and the collec-

tion as inputs and produces a ranked list of recommendations as output. RS engines range from typical IR engines to exclusively RS engines such as Collaborative Filtering. The main difference between IR systems and RS systems is that of focus: while IR systems focus on providing relevant ranked lists in response to a query, RS systems focus on predicting user preferences and providing a personalized list of recommendations based on history, trend or other clues.

Operationalization of each of the components of a recommender system vary immensely. Users can be represented in different ways; collections can come in several shapes and forms; and recommendation engines are implemented in diverse ways. Each choice of operationalization has its weaknesses and strengths, opportunities and challenges, and advantages and disadvantages. To assess the quality of a recommender system, an aggregate metric of quality is usually used.

The choices of operationalization in each of the components form a pipeline. A choice at any stage of the pipeline has an impact on subsequent stages. As such, an aggregate measure of the effectiveness of a recommender system may not show the effectiveness of choices made at different stages of the recommender system's pipeline. It may hide the weaknesses and/or strengths of choices.

In this thesis, we zoom in on key stages of the recommender system pipeline and investigate the impact of the choices made in a particular stage over the overall performance of a recommender system. For example, in Chapter 4, we investigate the impact of the interplay between feature sets and machine learning algorithms on the later stages of a recommender system. In Chapter 5, we conduct an in-depth study into filtering a collection, a very common initial stage in recommender system pipelines, and in Chapter 8, we examine the factors, patterns and inconsistencies in the evaluation of recommender systems. In particular, we show the challenges of evaluating recommender systems in different settings (offline and online using A/B testing), advise precautions and suggest solutions.

Last but not least, we go into an area where the impact of recommender systems has become a hot topic of debate in academia, industry and society: the debate on the impact of recommender systems, either as systems mitigating the information overload problem, or isolating users in filter bubbles. This debate presupposes that RSs are either under- or overpersonalizing recommendations. We take a user-centric perspective and propose a method for quantifying the degree of personalization. The method can help both industry and academia to quantify, objectively, this aspect of a recommender system from a user-engagement point of view.

This thesis' contributions are dotted around the entire recommendation pipeline involving users, collection and algorithms. We start with an exploration of approaches for Cumulative Citation Recommendation, a special recommendation task that aims to ease the curation and maintenance of knowledge bases such as Wikipedia. Following that, the thesis investigates the interplay between feature selection and choice of machine learning algorithm, and the impact on subsequently arrived conclusions. In the user and collection domains, the thesis investigates the initial stage of filtering a collection where different ways of user and entity representations are investigated, and their impact on later stages of a recommender system's pipeline is quantified. The thesis then investigates the evaluation of recommender systems from multiple dimensions: offline-online evaluations, randomness in A/B tests, and performance across time. Finally, the thesis deals with quantifying the impact of person-

alized content differentiation with a view to quantifying the degree of personalization of a recommender system.

1.1. Research Outline, Themes and Questions

This thesis investigates the recommendation pipeline by zooming in on specific stages to better understand the interactions of choices and components in the overall system performance. It does this under three themes: Cumulative Citation Recommendation (CCR), News Recommendation and Quantifying the Impact of Recommender Systems. CCR is the task of filtering and recommending citation-worthy documents from a collection so that curators of Knowledge Bases (KBs) such as Wikipedia can update the KBs. This task is important given the importance of keeping KBs up-to-date as a source of information and the need to filter the web for new citation-worthy documents to be used for updating the KBs.

News media play a central role in a democratic society [1]. News recommenders sort, select and rank news items, therefore taking over journalistic roles [2, 3]. The evaluation and impact of news recommendation are therefore important tasks encompassing many issues such as what should we measure and how, and what normative frameworks should we employ to measure the effectiveness of news recommendation. The themes of "News Recommendation", and "Quantifying an Impact of Recommender Systems" attempt to address some issues in news recommendation.

Below, under each of the themes, are the research questions that have been pursued.

1.1.1. Theme I: Cumulative Citation Recommendation

Under this theme, we pursue four research questions. Given a stream of documents and a set of pre-selected knowledge Base (KB) entities, CCR is defined as the task of filtering, ranking and recommending items or documents according to the relevance—citation-worthiness—of the documents to the target KB entity profile so that a KB curator can use the items or documents to update the entity's KB entry. This task is important as a result of 1) the importance, as an encyclopedic reference, of Knowledge Bases such as Wikipedia, 2) the number of new documents (information items) that appear on the Internet, 3) the number of KB entities that KB curators must deal with and 4) the small number of volunteers that work to update the KB entities. As a result of these factors, the time lag between a citation-worthy document appearing on the Internet and its inclusion in a pertinent KB entity can be large [4, 5]. In this theme, as part of the TREC Knowledge Base Acceleration benchmarking initiative where many teams took part in testing their best algorithms, we opted for a simple approach where entities are represented by an expanded set of labels. We explored the following research questions. The first is on the assessment of string-matching approaches to the CCR task and how that fares in comparison to other more sophisticated approaches. We asked the following research questions.

RQ1 How do simple string-matching approaches to the CCR task perform?

We attempt to answer this question in Chapter 3 where we employ the string-matching approach where entities are represented by a rich set of labels. Participation in the TREC KBA benchmarking task allowed us to compare our approach with the approaches of other participants. We present the factors that affect performance in a string-matching approach.

1

Following and also inspired by our finding that the string-matching approach was one of the best-performing ones in this task, we try to combine the best of two worlds: our effective features and then state-of-the-art machine learning algorithms. We asked the following research question.

RQ2 Does the use of the rich entity representations from our string-matching approach with machine learning approaches result in improved performances?

We answer RQ2 research question in Chapter 3, where we also compare the approach with the string-matching approach. Following this, we engage in the study of the impact of the interplay of choices in the recommendation pipeline. Machine learning approaches to the KBA-CCR task are complex involving multi-stage pipelines. We are interested in the impact of choices in one stage on the subsequent stages and on the overall performance. Recommender systems are usually evaluated based on only their overall performance and the best performing ones are adopted as the state-of-the-art. This, however, hides the weaknesses and strengths of choices at different stages of the pipeline. A recommender system that performs well in some stages of the pipeline might end up having a bad overall performance score due to choices on other stages of the pipeline, or vice versa. This means there might be gains that are overlooked or weaknesses that can be avoided by making other choices at a particular stage. We zoom in on some particular stages of the KBA-CCR task to investigate the impact of choices on subsequent stages of the pipeline and the overall performance. In particular, we ask the following research question.

RQ3 How does the interplay between the selection of features and the choice of algorithms affect performance?

RQ4 How big is the impact of the initial task of filtering in the KBA-CCR overall performance, and what makes documents unfilterable?

Chapter 4 answers RQ3 by investigating the impact of the choice of feature sets in two state-of-the-art machine learning algorithms. Chapter 5 answers RQ4 by thoroughly investigating the filtering stage.

1.1.2. Theme II: News Recommendation

News recommendation has its own unique challenges due to temporal and geographical factors, and due to the fact that items and users are in constant flux. In this theme, we explore several interrelated research questions on the factors affecting news recommendation and consumption, the challenges involved in the evaluation of news recommender systems and comparisons of evaluations and platforms. We investigate the different research questions under different sub-themes, which we presented below. The research work under this theme was done in a news recommendation evaluation initiative called CLEF NEWSREEL [6, 7] that provided a platform for the testing of news recommender algorithms.

Investigating the Role of Geography in News Consumption:
Under this investigation, we ask the following research question.

RQ5 What is the impact of geographical proximity on the consumption of news?

In Chapter 7, we seek to answer RQ5 by investigating the relationship between the geographical focus of online news portals and the geographical area of users in two online news portals. We then investigate the comparison of news recommender systems in a real-world setting. We incorporate the geographical relationship between news items and users from the previous question into our algorithms. Specifically, we ask the following research question and seek to answer it in the same chapter.

RQ6 What are the patterns of news recommender performance in real-world news recommendation, and does the incorporation of geographical information improve performance?

We attempt to answer RQ6 by deploying several interrelated algorithms, one of which incorporates geographical information into its selection of recommendation items. To investigate empirically the effect of non-algorithmic factors in A/B testing, we deployed two instances of the same algorithm.

Multidimensional Investigation of News Recommendation

In Chapter 8, we seek answers to 3 interrelated research questions. The goal here is to better understand the challenges, validity and fairness, and consistency of news recommender system evaluations. Building on our observation of the performance of the two instances of the same algorithm in Chapter 7, we begin this chapter with the following research question.

RQ7 What are the causes of "random" performance differences in real-life news recommendation, and how can we quantify the extent of random differences?

We answer RQ7 by presenting and classifying the possible causes and proposing a way to quantify "random" and idiosyncratic performance differences. After this, we proceed to investigate the validity and consistency of news recommender evaluations in offline and online settings, and across time. We ask the following research question:

RQ8 How do news recommender system performances compare offline, online and across time periods?

Finally, we complete our investigation into the evaluation of news recommender systems by appraising the platform where our news recommender algorithms were evaluated. The research question we asked is:

RQ9 What are the participant perspectives on the evaluation of their recommender systems in CLEF NEWSREEL?

1.1.3. Theme III: Measuring Recommender System Personalization

In this theme, we attempt to quantify a recommender system's personalization success from a user perspective. Recommender systems are under continuous debate regarding their possible role in creating filter bubbles. Others argue that recommender systems are useful for mitigating information overload, which is a fact of life with the explosion of content on the Internet. We touch upon this topic, the different aims and ways of measuring personalization and subsequently propose a metric for quantifying the degree of personalization success from a user-centric perspective. We specifically ask the following research question.

1

RQ10 Can we quantify the degree of over- or underpersonalization by a recommender system from a user-centric point of view?

We answer RQ10 in Chapter 10. The motivation, need and description for quantifying the under- or overpersonalization is provided and a demonstration of the method on real-world datasets is presented.

1.2. Thesis Overview

The thesis is organized in three parts. The overall thesis structure can be seen in Figure 1.1. The first part explores parts of the recommendation pipeline under the theme of Cumulative Citation Recommendation, which is the task of filtering, ranking and recommending citation-worthy and relevant items to Knowledge Base curators. The second part investigates different factors and aspects of news recommendation and their evaluation, and the third part attempts to propose an evaluation metric that can be applied to evaluate an aspect of recommendation effectiveness in both themes.

1.3. Publications

The following are the publications upon which this thesis is based.

Chapter 3

Samur Araújo, **Gebrekirstos G. Gebremeskel**, Jiyin He, Corrado Boscarino, Arjen P. de Vries. CWI at TREC 2012, KBA track and Session Track. In Proceedings of TREC '12, 2013. The "KBA Track" is the author's contribution.

Alejandro Bellogín, **Gebrekirstos G. Gebremeskel**, Jiyin He, Alan Said, Thaer Samar, Arjen P. de Vries, Jimmy Lin, Jeroen B. P. Vuurens. CWI and TU Delft Notebook TREC 2013: Contextual Suggestion, Federated Web Search, KBA, and Web Tracks. In Proceedings of TREC'13, 2014. The "KBA" part is the author's contribution.

Chapter 4

Gebrekirstos G. Gebremeskel, Jiyin He, Arjen P. de Vries, Jimmy Lin. Cumulative Citation Recommendation: A Feature-Aware Comparison of Approaches. In: 25th International Workshop on Database and Expert Systems Applications, DEXA '14, pp. 193–197 (2014)

Chapter 5

Gebrekirstos G. Gebremeskel, Arjen P. de Vries. Entity-Centric Stream Filtering and Ranking: Filtering and Unfilterable Documents. in European Conference on Information Retrieval(Springer, 2015) pp. 303–314.

Chapter 7

Gebrekirstos G. Gebremeskel and Arjen P. de Vries. The Role of Geographic Information in News Consumption. In Proceedings of the 24th International Conference on World Wide Web (WWW '15 Companion). pp. 755-760.

Chapter 8

Gebrekirstos G. Gebremeskel and Arjen P. de Vries. The Degree of Randomness in a Live Recommender Systems Evaluation. Conference and Labs of the Evaluation Forum (2015).

Gebrekirstos G. Gebremeskel and Arjen P. de Vries. Recommender Systems Evaluations: Offline, Online, Time and A/A Test. In Working Notes of the 7th International Conference of the CLEF Initiative, Evora, Portugal. CEUR Workshop Proceedings, 2016

Gebrekirstos G. Gebremeskel, Arjen P. de Vries. Random Performance Differences Between Online Recommender System Algorithms. In Fuhr N. et al. (eds) Experimental IR Meets Multilinguality, Multimodality, and Interaction. CLEF 2016. Lecture Notes in Computer Science.

Benjamin Kille, Andreas Lommatzsch, **Gebrekirstos G. Gebremeskel**, Frank Hopfgartner, Martha A. Larson, Jonas Seiler, Davide Malagoli, András Serény, Torben Brodt, Arjen P. de Vries. Overview of NewsREEL'16: Multidimensional Evaluation of Real-Time Stream-Recommendation Algorithms. In CLEF 2016: 7th Conference and Labs of the Evaluation Forum, June 2016: 311-331

Chapter 10

Gebrekirstos G. Gebremeskel, Arjen P. de Vries. Pull–Push: A Measure of Over- or Under-personalization in Recommendation. International Journal of Data Science and Analytics (2022). https://doi.org/10.1007/s41060-022-00354-9

1

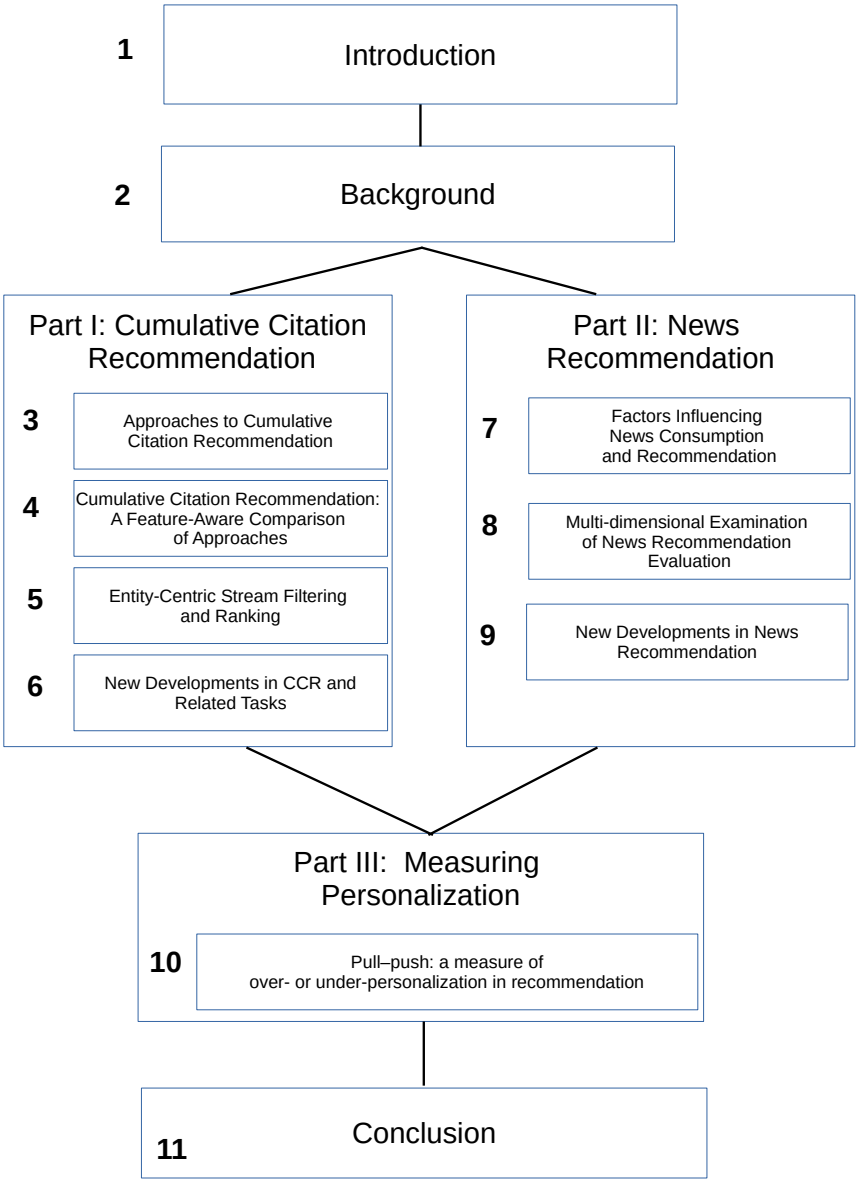


Figure 1.1: Overview of the thesis structure. It consists of two research themes (parts) that can be read independently and a third part that can be read after either of the two.

2

Background

We provide the background context and information for the research themes here. For more (new) related work and how they relate to or affect our findings and results, please have a look at "New Developments in CCR and Related Tasks" (Chapter 6) for the theme of Cumulative Citation Recommendation, and "New Developments in News Recommendation" (Chapter 9) for the theme of News Recommendation.

Recommender systems came into existence with the explosion of content on the Internet [8]. They are supposed, among other things, to address the problem of information overload [9]. The task of a recommender system is to match users with items of their interest. More specifically, the recommender system must select the items of interest out of a collection and rank them for the user. Recommender systems are assumed to have value to both the customer and the provider. To customers, they can help them find interesting (relevant) items, save time and decision effort, discover new things or keep them engaged. To the provider, they are assumed to increase sales, engagement and conversion, provide opportunities for promotion and persuasion [9].

Today, recommender system are in use everywhere on the Internet, and they are integral parts of companies' revenue and services. E-commerce sites, hotel reservation sites, news aggregators, social media sites, search engines (for computational advertising) and news portals use recommender systems. Recommendations can be displayed in different ways. For example, while the recommendations in e-commerce website Amazon in Figure 2.1, in hotel reservation website Booking.com in Figure 2.2 and in the New York Times in Figure 2.4 represent the more general form of recommendation display, which is at the bottom following the item being viewed, the New York Times recommendations in Figure 2.3 represent an "in Text" display.

Recommender systems are implemented in different ways depending on their purpose, availability of interaction history, and suitability for the task in question. The most common methods are Content-based recommenders, Knowledge-based recommenders and Collaborative Filtering. Content-based recommender systems rely on a description of the user profile and the items. One of the research themes we took part in, TREC KBA [5, 10], assumes the use of Knowledge-based recommender systems. TREC KBA provides a set of

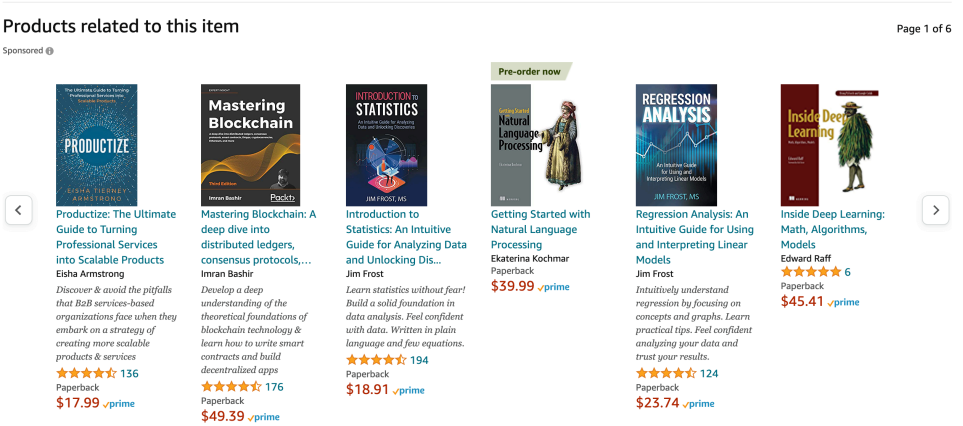


Figure 2.1: Book recommendations in amazon.com for viewing “Recommender Systems: An Introduction 1st Edition”.

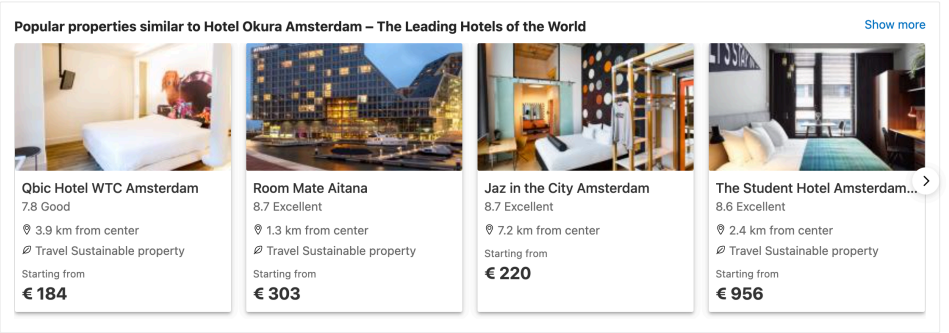


Figure 2.2: Accommodation recommendations in booking.com for viewing “Hotel Okura Amsterdam”.

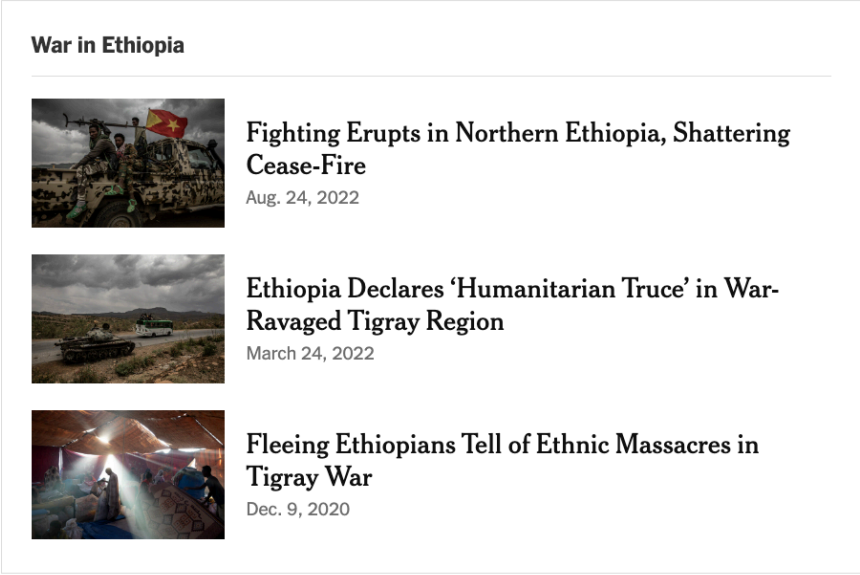


Figure 2.3: In-text News recommendations in the New York Times for viewing "Ethiopian Airstrike Hits Kindergarten as Fighting Spreads in Tigray".

Knowledge Base entities from Wikipedia or Twitter, and a stream of documents. The task is to filter, rank and recommend items that have citation-worthy or relevant information for the KB entities of interest.

The most popular recommender systems use Collaborative Filtering (CF), which can be user-based or item-based. CF uses history of users or items' similarity to recommend new items [11]. CF relies on (sufficient) availability of interaction history, and users and items must be identified by a persistent name or id. This method is suitable for websites such as Amazon where users and items can be identified, and enough interaction history is collected. This method is, however, not suited for news recommendation where users browse anonymously, items have a short lifespan and interaction history is limited [12]. These factors make news recommendation different and call for a different approach. Therefore, in this theme, we explore different approaches and investigate the challenges of evaluating news recommender systems.

Recommender systems may have intended or unintended consequences. One controversial topic that has arisen in the public debate about recommender systems is the idea of filter bubbles. Critics accuse (personalized) recommender systems of segregating society and insulating users in filter bubbles [13]. Others argue that recommenders have a useful purpose in mitigating the problems of information overload [9]. A part of this debate can be conceptualized as whether recommender systems are over- or underpersonalizing. To answer this, we propose a metric for quantifying the effect of a recommender system in this regard. In the following sections, we describe the background information for the three themes addressed

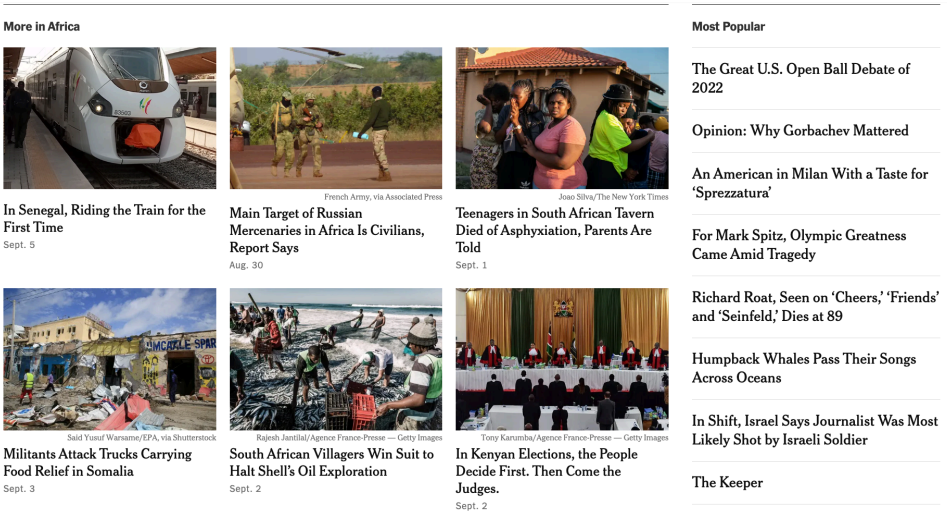


Figure 2.4: General News recommendations in the New York Times displayed at the bottom for viewing the news item "Ethiopian Airstrike Hits Kindergarten as Fighting Spreads in Tigray".

in this dissertation.

2.1. Cumulative Citation Recommendation

Knowledge Bases (KBs) have gained importance and popularity. KBs are not only useful for providing general encyclopedic reference for humans, but also for supporting information seeking tasks such as information retrieval, semantic search and entity linking [14]. They are also useful for various knowledge extraction and mining tasks [4]. KBs provide or strive to provide key information about entities, their properties, attributes and their relationships to other entities. Large KBs, such as Wikipedia, contain millions of entities and billions of facts about those entities [14]. As KBs are curated and maintained by a small number of people, the huge number of entities and the facts about them pose a great challenge for maintenance. Especially challenging is the fact that KBs are always in need of updating the information about entities as new facts about them keep appearing.

One problem with human curation of KBs is the time lag between the appearance of pertinent information on the Internet and its inclusion in the relevant entity profile, for example in its Wikipedia profile. An analysis of 60,000 web pages cited by Wikipedia in the Living_people category shows a median of 356 days time lag and a long tail [5] (see the plot in Figure 2.5).

The challenges in updating KBs have inspired initiatives that aim to ease the updating process. One such initiative is the Text Analysis Conference's Knowledge Base Population track (TAC KBP), whose aim is to promote research in discovering and extracting facts about entities and to populate a KB entry with these facts [15, 16]. TAC KBP is essentially, given a collection and a set of entities, the task to discover and extract relevant facts from the collection about the entities. The official TAC KBP website lists five tracks: 1) Cold

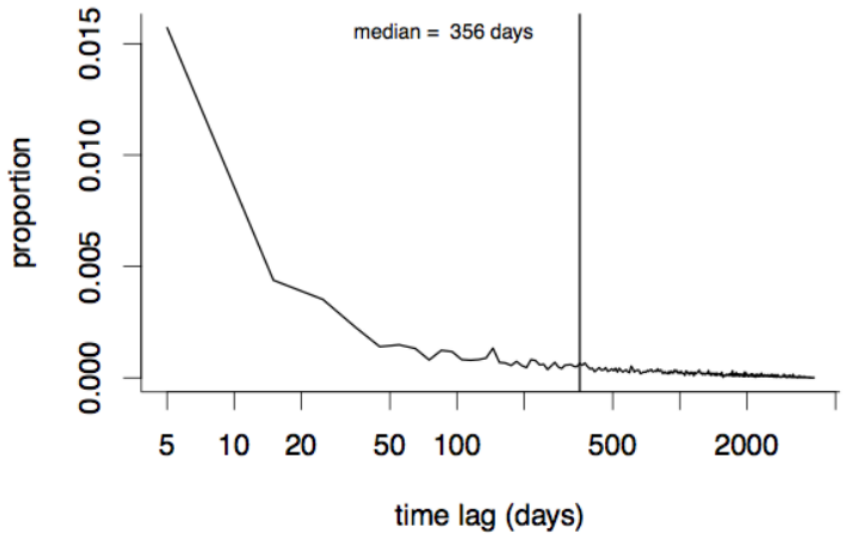


Figure 2.5: A figure (courtesy of TREC KBA 2012) showing an analysis for a sample of Wikipedia-cited web pages of time lag between new relevant information appearing on the Internet and being updated on the relevant Wikipedia page. The median is 356 days.

Start KB (build a knowledge base from scratch using a given document collection and a pre-defined KB schema), 2) Entity Discovery and Linking (extract entity mentions from a source collection and link them to a reference KB; and where KB entries do not exist, cluster mentions for those entities), 3) Slot Filling (search a document collection to fill in values for predefined slots (attributes) for a given entity), 4) Event (extract KB-worthy information about events) and 5) Belief/Sentiment (detect belief and sentiment of an entity toward another entity, relation, or event).

The TAC KBP initiative attempts to tackle several aspects of the discovery and extraction of relevant information about KB entities from a collection of text. The ambition is to replace human curators. A related initiative was the TREC KBA track. This track's aim is to promote research on filtering a time-ordered corpus for documents that warrant an update/edit of an entity profile in a KB for a predefined list of KB entities [10]. TREC KBA's central interest was CCR, defined as: given a stream of documents and a predefined set of KB entities, filter, rank and recommend citation-worthy documents. In other words, systems are required to make a distinction between documents that have new and citation-worthy information and those that have relevant content but do not warrant an update or editing of the KB profile. TREC KBA, unlike TAC KBP, does not aim to replace the human curator, but instead to present the human curator with ranked list of documents that necessitate a change in a KB entity's KB profile.

TREC KBA started in 2012, and continued with some changes and additions of tasks every year. We participated in 2012 and 2013. In both 2012 and 2013, the main task was to

filter a large time-ordered stream corpus for documents that are relevant for predefined KB entities. The task required systems to distinguish between centrally relevant (documents that have citation-worthy information) and relevant documents (documents that contain relevant information, but do not warrant the update of an entity's KB profile) [5, 10]. While the main task remained more or less the same in 2012 and 2013, the size and type of the corpus and the predefined list of KB entities either changed and/or were expanded.

TREC KBA participants used different approaches ranging from simple string matching to machine learning approaches. Systems have been evaluated against assessor judgments, using an overall F-score. TREC KBA systems, however, involve a lot of components and such overall measures may not be sufficiently informative for participants when they want to see where their systems strength or weaknesses lie. In our research, we attempt to zoom in on the components of a TREC KBA pipeline with the goal to understand the interplay between the components and their impact on the overall performance. Our research in these areas are attempts to draw attention to the components of a multi-component system in order to gain insights on the interplay between choices at different stages and the overall performance of systems.

2.2. News Recommendation

The Internet has had a massive impact on news delivery and consumption. Lower barriers to entry, possibility of reaching wider audiences, and lower cost of production and distribution have resulted in the migration of traditional print media to online news portals and the emergence of new Internet-native news portals. Consequently, the Internet has democratized the news landscape. Today, online news consumption constitutes a major part, if not the main, of delivering and receiving information. According to Pew Research Center[1], 86% of US adults access online news in one or another format.

In traditional print media, a few companies used to play the gate-keeping function. With the coming of the Internet, this function of companies had to give way to other forms of gate-keeping. Initially intended to help the user overcome the problem of information overload and find information of their interest, news aggregators such as Google News, Yahoo News and news portals use recommender algorithms to select and present items to the user, thus playing the role of a new gate-keeping function. Algorithmic news recommendation has become a fact of life. News portals and news aggregators present personalized lists in their home pages, display recommendations either in a widget or at the bottom of a news item page.

News recommendation has been of interest in the area of civic discourse, industry and academia. In the area of civic discourse, a hot debate has been going on on whether recommender systems are isolating users in filter bubbles, and in industry, recommender systems have evolved to be very diverse in algorithms, the features they use, speed and complexity [17]. In academic research, news recommendation has attracted much attention [18]. Disregarding interest by individual persons and universities, two academic initiatives that involve the participation of many academic institutions show the scale of interest in news recommendation. One of these is the TREC News Track [19], a NIST TREC initiative to foster research in news recommendation. The second one is the CLEF NewsREEL initiative [6, 7],

[1]https://www.pewresearch.org/journalism/fact-sheet/news-platform-fact-sheet/, accessed on 10 October 2024

a similar initiative that focuses on practical research into news recommendation to improve our understanding of the factors that affect news recommendation.

Research in news recommendation has focused on recommendation algorithms, factors that influence news consumption and/or clicks, evaluation of recommender systems, and non-algorithmic factors in news recommendation [6, 7, 20]. Our research in news recommendation was largely conducted by participating in NewsREEL Evaluation campaigns, where algorithms were plugged in to a platform to deliver live recommendations to actual users visiting information portals. In the NewsREEL News evaluation campaign, communication between recommendation providers and users was made possible by a platform called the Open Recommendation Platform (ORP) [21] which delivered recommendation to users and user interactions to the recommendation providers to build their models.

Online news recommendation poses its own unique challenges. Latency, the dynamic nature of users and items, the short life span of items, the anonymous natures of users, algorithmic and non-algorithmic factors play roles [7, 21]. In our study, we focused on studying factors that impact news recommendation, offline/online evaluations, A/A testing (running two instances of the same algorithm), the challenges of random artifacts in news recommendation evaluation and an appraisal of the evaluation platform.

2.3. Measuring Recommender System Personalization

Recommender systems are used for different purposes and are measured by different success criteria. Some are intended to increase engagement, others are intended to increase conversion (users buying products) and yet others provide serendipitous recommendations. As a result of this, there exist no holistic evaluation scenario [22]. In our study, we focused on quantifying an RS's personalization success from a user perspective. The information overload - filter bubble issue has been a talking point in media, academia and industry. Some argue that recommender systems help alleviate the information overload problem that came to exist as a result of the explosion of content on the Internet [17, 23]. Others argue that recommender systems have ushered in a situation where users are isolated in filter bubbles of only partly their own making [24, 25]. Implicit in these issues is an assumption that RS's are either over- or underpersonalizing.

In our study here, we took a user-centric perspective and attempted to provide a metric for quantifying the degree of a RS's under- or overpersonalization from the user perspective. Using several datasets, we demonstrated how the metric can be used and how the results can be interpreted. We believe the metric helps to partially concretize the debate on the impact of RS's personalization and to encourage further initiatives to devise metrics in this area.

I

Cumulative Citation Recommendation

3

Approaches to Cumulative Citation Recommendation

This chapter is based on our participation in the TREC Knowledge Base Acceleration Track in 2012 and 2013. In both years, we participated in the task of Cumulative Citation Recommendation (CCR), which is defined as filtering a time-ordered stream (or corpus) for documents that are citation-worthy to a predefined list of Knowledge Base entities. The goal of this task is to ease the manual tasks of Knowledge Base curation and maintenance. The "cumulative" part refers to the potential successive and incremental updating of Knowledge Base (KB) entities with citation-worthy information recommended by a CCR system. While the CCR task setup remained the same between both years, the set of entities and documents changed in the years. In 2012, there were only Wikipedia entities, but in 2013 there were Wikipedia and Twitter entities. The size and content of the stream documents also changed. In 2013, there was content from social media sites (eg. Twitter). In both years, the entities were English Wikipedia Entities and the stream documents were also English.

For TREC 2012, the KB entities consisted of a set of 29 Wikipedia entities (27 people and 2 organizations). The stream corpus consisted of 400M documents. For TREC 2013, a set of 141 Wikipedia and Twitter entities (98 people, 19 organizations, and 24 facilities) were chosen. In both years, a set of relevance judgments were provided for training and testing purposes.

The relevance judgments assign a document-entity pair one of the labels of central, relevant, neutral or garbage. The central and relevant labels are alternatively referred to, especially in 2013, by vital and useful, respectively. Assessors were instructed to "use the Wikipedia article to identify (disambiguate) the entity, and then imagine forgetting all info in the Wikipedia article and asking whether the text provides any information about the entity" [5]. A document labeled with one of the labels can be mentioning or non-mentioning. A mention means a document explicitly mentions the target entity, such as full name, partial name, nickname, pseudonym, title, or stage name. A document can mention an entity and yet be labeled neutral. Similarly, a document that does not mention an entity can still be relevant. TREC KBA defines the relevance labels as follows.

Garbage: Not relevant, e.g. spam.

Neutral: Not relevant, i.e. no info could be deduced about the entity, e.g., entity name used in product name, or only pertains to community of target such that no information could be learned about entity, although you can see how an automatic algorithm might have thought it was relevant.

Relevant: Relates indirectly, e.g., tangential with substantive implications, or topics or events of likely impact on entity.

Central: Relates directly to target such that you would cite it in the Wikipedia article for this entity, e.g. entity is a central figure in topics/events.

To accomplish the CCR task, we employed a string matching approach in 2012 and a machine learning approach in 2013. Our machine learning approach in 2013 was inspired by our well-performing rich string representations of our approach in 2012. We set out to incorporate the string representations of 2012 with machine learning approaches that were also shown to perform well in other participants' works. Although performances did not meet our expectations in 2013, we obtained important insights. Our approach targets the central and relevant labels. One of our methods was one of the top-performing approaches in the KBA track evaluations.

3.1. String Matching As an Approach to CCR

TREC KBA provided a collection of stream documents, a set of Wikipedia entities of interest and a set of relevance judgments. For each pair of document and Wikipedia entity, relevance judgments give one of the labels of central (citation-worthy), relevant, neutral or garbage. The task is then for each document-Wikipedia Entity pair, present a ranked list of document-Wikipedia Entity for unjudged document-Wikipedia Entity pairs.

Initial assessor results reported that 4% of the Wikipedia citations did not mention the Wikipedia entities they are cited by. We thought there could be more documents in the stream that do not mention the Wikipedia entities by name and yet are relevant. Capturing relevant documents that do not mention entities by name might seem attractive. That would also imply a need for a way of entity representation that accounts for the evolution of the entity as documents that are non-mentioning and yet central (relevant) are found. Statistics from the few weeks' relevance judgments changed the way we see the task. As it can be seen from Table 3.1, out of all non-mentioning documents, only 0.4% are relevant, 0% are central. So we did not see, from a performance perspective, a point in concentrating our efforts on detecting non-mentioning-yet-relevant stream documents. Instead, we decided to focus only on mentioning and relevant or/and central labels.

3.1.1. Our approach

Out of all the mentioning documents, 23.8% are garbage, 35.3% are garbage or neutral. Now, the challenge is not how to filter non-mentioning-yet relevant, but how to exclude mentioning-yet-non-relevant, i.e. garbage and neutral. Another challenge is that the entities are ambiguous in the sense that two entities can have the same or nearly the same name, and thus the same representation. This observation informed our next choices of approaches.

Table 3.1: Statistics from a few weeks assessor judgments on mentioning and not-mentioning documents for the different relevance labels. Alternatively, the "central" label is also called "vital", and the relevant "useful".

	garbage	neutral	relevant	central	total
Mention	7991	3862	13971	7806	33630
Not mentioning	15367	163	61	0	15591

We sought approaches that can, at least, solve one of the two problems. We decided to use a resource called Google Cross Lingual Dictionary (GCLD) [26], which helps resolve the ambiguity in representation by assigning probabilities for string- Wikipedia concept and Wikipedia concept-string mappings.

This means to first represent the queries (which are the Wikipedia entities in our case) using the relevant strings in GCLD resources and then to query the stream. After finding a match of the strings for an entity in the stream, we use the probabilities to give a confidence score. All our approaches revolve around the choices of entity representation, the scoring function to measure confidence and scaling functions.

3.1.2. Representation

Representing the Wikipedia entities (the queries) and the stream documents in some way is mandatory. At first, we thought we can represent the streams in terms of n-gram tokens thereby reducing the size of the corpus, but then that would confine us to only approaches that can consume the tokens. So we left stream documents largely unchanged except for minor changes from preprocessing. We needed, however, to represent the Wikipedia entities in some way. All our approaches used a different entity representation and that is the main component. The components of our approaches are entity representation, string matching, scoring and, to some extent, scaling functions.

In the approaches that use the GCLD, Wikipedia entities are represented by strings and those strings are used to query the stream. If a matching string is found, the document is labelled as relevant and/or central to a degree provided by a scoring function. We have experimented with many variations of scoring functions, thresholds and scaling functions.

Google Cross Lingual Dictionary (GCLD) Approach

The GCLD resource associates, with probability scores, strings of natural language text with English Wikipedia entities (also called concepts or URLs). The resource assigns empirical probability distributions to a string being used to refer to a Wikipedia entity and vice versa [26]. Our approach here is to represent the Wikipedia entities by the strings in the dictionary and to use the representation to filter from the stream documents that are central or/and relevant. The probabilities are used to give a confidence score for the relevance of a document for a Wikipedia entity. The dictionary has other statistical information that can be used in different ways to improve performance, and we have tried to experiment with some and examined their effects. Below we will detail the dictionary, the approach we used, the experiments we did and discuss the results and draw conclusions.

The dictionary is bidirectional in the sense that it provides a mapping from free-form natural language strings to concepts and vice versa. The strings are gathered from anchor texts

to all Wikipedia pages and from the English Wikipedia titles. This means the strings include anchor texts from inter-Wikipedia linking, and anchor texts to non-English Wikipedia articles. The strength of association between strings and concepts is quantified by conditional probabilities.

Let $s \in S$ be a string and let $e \in E$ be an English Wikipedia entity, represented by a unique URL. $l(s,e)$ is a link between s and e where s is used as an anchor in a link to a Wikipedia entity e. $\#l(s,e)$ is the total number of hyperlinks into a Wikipedia article e using anchor text s. $\sum_{e \in E} l(s,e)$ is the total number of links into Wikipedia pages that use s as an anchor and $\sum_{s \in S} l(s,e)$ is the total number of links to a Wikipedia article e. Based on this, the dictionary defines two probabilities : $P(e|s)$ for strings to concepts and $P(s|e)$ for concepts to strings as follows.

$$P(e|s) = \frac{\#l(s,e)}{\sum_{e \in E} l(s,e)} \tag{3.1}$$

$$P(s|e) = \frac{\#l(s,e)}{\sum_{s' \in S} l(s',e)} \tag{3.2}$$

Equation 3.1 tells whether a string is ever used as an anchor text to a certain Wikipedia entity, and if it does, it gives the probability. By this, it disambiguates the string by distributing the probability mass over the different Wikipedia entities according to how often it is used as an anchor to each of them. Equation 3.2 quantifies how important as an anchor a certain string is in comparison to other strings that also point to the same entity e. All strings that can be used as anchors to a certain Wikipedia article are co-referents, and the second formula measures the relative strength with which a co-referent refers to a Wikipedia article.

An alternative way to look at the two Equations in Equation 3.1 and Equation 3.2 is to interpret them analogous to the tf-idf concept. Analogous to the tf-idf concept, a document is the number of links pointing to a Wikipedia article. The table in 3.2 relates the elements of the Equations with those of the tf-idf.

One can think that the first formula is like the term frequency normalized by the number of terms and the second formula is a modified idf, i.e. it uses the number of links into a document instead of the number of documents in a collection, and normalizes it by the number of all links having anchor s.

3.1.3. Discussion

We conducted many experiments by varying dictionary strings for representation, probabilities for scoring, and thresholds for selecting strings. The algorithm is string matching, i.e. once the Wikipedia entities are represented with our choice of set of strings, we query each document of the stream to see if it has a match for the entity representations. If there is a match, we give the document-entity pair a confidence score computed based on the probabilities. When more than one element of the set of strings for an entity are matched, we take either the average or the maximum of the probabilities of the matched strings. Our measures were recall, precision, and F-measure against relevance cut-off. But to distinguish between two approaches, we mainly looked at F-measure.

Table 3.2: Analogy between TF-IDF and GCLD concepts.

tf-idf	GCLD
t =term	l(s,e) - an instance of a link between anchor s and Wikipedia entity e
tf = term-frequency	#l(s,e) - the total number of hyperlinks to a Wikipedia article having s as anchor
d	$\sum_{s \in S} l(s,e)$-all links to a Wikipedia article
df_t	The total number of links that contain s, $\#\sum_{s \in S} l(s,e)$ that contain particular l(s,e), $lf_{l(s,e)}$
N	$\#\sum_{s \in S} l(s,e)$ - the collection, the number of Wikipedia entities in this case
$idf_t = log \frac{N}{df_t}$	$idf_{l(s,e)} = \frac{\#l(s,e)}{\sum_{s \in S} l(s,e)}$

Our first experiment was with probabilities given by $P(e|s)$. We lowercased all the string representations of the Wikipedia entities and the stream documents. When two strings are lowercased to the same form, we assign the form that we keep the higher probability. We also stripped punctuation and white spaces. We did experiments with strings that come only from non-Wikipedia pages, and all strings to English or corresponding non-English Wikipedia pages. We compared the results on F-measure and the later representation performed better. The reason for increment in F-measure was because of an increase in recall. And the increase in recall is due to the additional strings. Using the average of the probabilities of the matching strings performed worse than the maximum. Next, we experimented by setting different thresholds on probabilities in order to select strings that have higher probabilities. We tried thresholds 0.01, 0.001, 0.0001. However, the performance did not improve significantly. In fact, in the case of threshold 0.01, performance dropped significantly.

Our second experiment was in lowercasing and stripping punctuation. The dictionary strings that we used are not lowercased, i.e. “Nasim” and “NASIM” are considered different strings. The dictionary strings also contain punctuation and white spaces. We decided to experiment without lowercasing the entity representations and the stream documents. The performance was a big improvement over the lowercased and punctuation-stripped approach. It is not surprising that it is so since it better captures the capitalization which are a feature of proper nouns.

We were, however, not satisfied with the results, since the performance scores were still poor. Moreover, the confidence scores were very small and were very susceptible to scaling functions. $P(e|s)$ is like a tf, it never tells us how discriminative a string is to a certain Wikipedia entity with respect to other Wikipedia entities. $P(s|e)$ is the right probability to use for this purpose. $P(s|e)$ exposes the ambiguity in a string by distributing the probability mass over the entities it can be used as anchor in a link. There are many strings whose $P(s|e)$ probabilities were 1, which shows that the document (d) containing the string is highly probably relevant to the WP entity the string represents. And, indeed, experiments using these probabilities for scoring showed better performance. The use of $P(s|e)$ dis-

ambiguates ambiguous entities naturally. Varying thresholds for string selection and using averaging instead of maximum did not improve results significantly.

Our third experiment was therefore combining the two probabilities for scoring. When more than one string is matched, we multiply both probabilities first, and keep the maximum as a score. Our best scores on the relevant and central labels was obtained by this approach. Our main run submissions were from this approach. We also submitted runs using the same approach but with lowercased and punctuation-stripped. For both cases, we used two different simple per-entity scaling functions. First, we selected the maximum score per entity and use that to scale the document (d) and entity results as:

$$s_{scaled}(d,e) = \frac{s(d,e)}{s_{max}(e)} * 1000 \tag{3.3}$$

In our second scaling function, we used a threshold on maximum score per entity to discourage entities whose maximum score is less than 10. All of our runs use GCLD's strings and probabilities to represent the entities and search the documents for a match. They all use $P(e \mid s)$ multiplied by $P(s \mid e)$ for scoring. Using this combined scoring with and without lowercasing and stripping and the two scaling functions, we submitted the following four runs.

gcld1 uses Google GCLD's strings and probabilities to represent the entities and searches the documents for a match. It does not strip nor lowercase the strings and docs. The score is normalized by the highest score per entity and multiplied by 1000. However, entities whose max score is < 0.01 are normalized by 0.01 to discourage them from being equally competitive with other entities.

gcld3 uses GCLD's strings and probabilities to represent the entities and searches the documents for a match. It does not strip nor lowercase the strings and docs. The score is normalized by the highest score per entity and multiplied by 1000.

gcld_s1 uses GCLD's strings and probabilities to represent the entities and searches the documents for a match. It strips punctuation and lowercases the strings and docs. If many strings are found to match, the highest score is chosen as the score for the doc-entity pair. The score is then normalized by the highest score per entity and multiplied by 1000. Entities whose highest Score is < 0.01 are normalized by 0.01 to discourage them from being equally competitive with other entities.

gcld_s2 uses GCLD's strings and probabilities to represent the entities and searches the documents for a match. It strips punctuation and lowercases the strings and docs. If many strings are found to match, the highest score is chosen as the score for the doc-entity pair. The score is then normalized by the highest score per entity and multiplied by 1000.

Run Graphs and comparisons

Figures 3.1 and 3.2 show the performance on the test set of the two best performing variations of GCLD. Table 3.3 shows the highest score for each entity and the run that generated it.

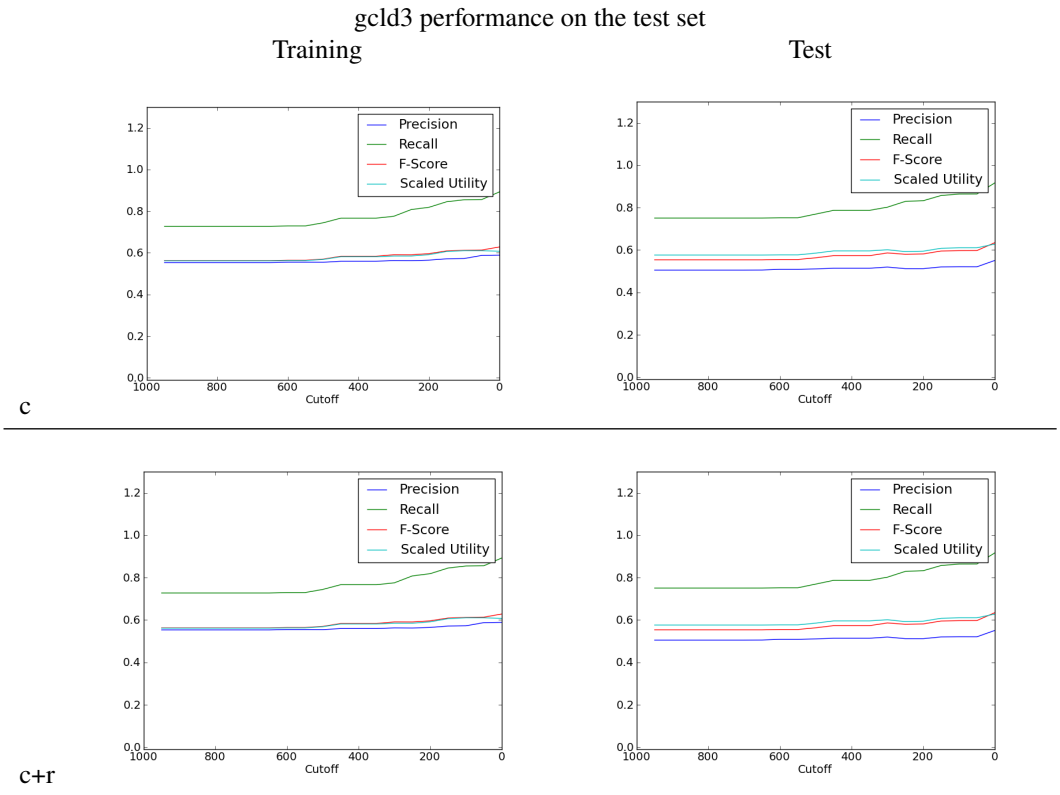


Figure 3.1: Results of GCLD no stripping normalized by highest score. Training results in the left column and testing results in the right column. Top row is central, bottom row is central+relevant.

As it can be seen in Table 3.3, the non-stripping, non-lowercasing runs perform better. This is not surprising since stripping and lowercasing can take away some of the characters that identify personal nouns. We also observe that in many cases the performance for central and central+relevant are the same. That is because our system does not do well at distinguishing between them.

Comparison to Other Participating Systems

We present the comparison of our system with other participating systems. The plots in Figure 3.3 and Figure 3.4 below are provided by TREC KBA. As can be seen from the plots our systems (gcld1, gcld3) are some of the top performers.

Our runs are in the CWI column, namely google_dic1 (gcld1), google_dic2 (gcld2), google_dic3 (gcld3), google_strip1 (gcld_s1), and google_strip2 (gcld_s2). We did not include google_dic2 in our analysis because it was submitted accidentally. Three of our runs performed very well in both macro-averaged F-score and highest F-score. In macro-averaging, scores for each entity are computed and then averaged across entities with equal weight per entity. The highest scores are based on computing the F-score after macro-

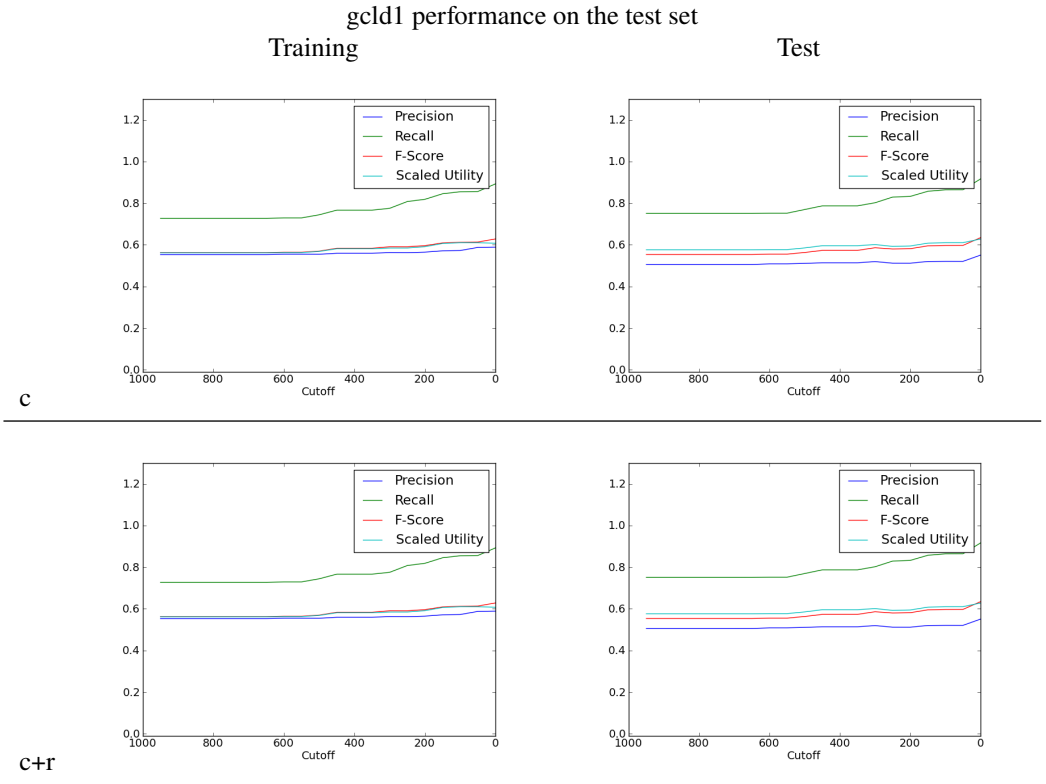


Figure 3.2: Results of GCLD with no stripping and normalized by highest score except those less than 0.01. Training results in the left column and testing results in the right column. Top row is central, bottom row is central+relevant.

averaging the precision and recall scores across the entities. Our runs are specially very good at targeting both the central + relevant labels. The approach is not good at distinguishing between central and relevant documents, as can be seen from how it has performed in the central bin, as compared to the central + relevant.

3.2. A Machine Learning Approach to CCR

Our participation in KBA 2013 was inspired by a desire to combine the best performing aspects of several approaches. In TREC KBA 2012, we experimented with string-matching for CCR [27]. We represented the Wikipedia entities with rich features from a resource called Google Cross Lingual Dictionary (GCLD) which is a mapping (with probability distributions) from strings to Wikipedia concepts and vice versa. This approach performed well in general, and it was very good at recall in particular. We noted also high-performing approaches from other TREC 2012 participants included an approach that used entity and related entity mentions [27, 28]. We used the then state-of-the-art machine learning approaches and a huge feature set for the CCR [29, 30]

Table 3.3: The highest maxF scores for each entity and the run that produced it (Results sorted by the maxF of the central column.)

Entities	Central		Relevant+central	
	maxF	Approach	maxF	Approach
Mario_Garnero	0.952	gcld1, gcld3	0.952	gcld1, gcld3
Ikuhisa_Minowa	0.612	gcld1, gcld3	0.612	gcld1 , gcld3
Satoshi_Ishii	0.604	gcld1, gcld3	0.604	gcld1, gcld3
Basic_Element_(company)	0.594	gcld1, gcld3	0.594	gcld1, gcld3
Boris_Berezovsky_(businessman)	0.544	gcld1, gcld3	0.544	gcld1, gcld3
Roustam_Tariko	0.466	gcld1, gcld3	0.466	gcld1, gcld3
Nassim_Nicholas_Taleb	0.463	gcld1, gcld3	0.463	gcld1, gcld3
William_D._Cohan	0.395	gcld1, gcld3	0.395	gcld1, gcld3
Annie_Laurie_Gaylor	0.392	gcld1, gcld3	0.392	gcld1, gcld3
Ruth_Rendell	0.386	gcld1, gcld3	0.386	gcld1, gcld3
Vladimir_Potanin	0.336	gcld1, gcld3	0.336	gcld1, gcld3
Frederick_M._Lawrence	0.333	gcld1, gcld3	0.333	gcld1, gcld3
Alex_Kapranos	0.298	gcld1, gcld3	0.298	gcld1, gcld3
James_McCartney	0.281	gcld1, gcld3	0.281	gcld1, gcld3
Darren_Rowse	0.270	gcld1, gcld3	0.270	gcld1, gcld3
Bill_Coen	0.231	gcld1, gcld3	0.231	gcld1, gcld3
Alexander_McCall_Smith	0.221	gcld1, gcld3	0.221	gcld1, gcld3
Aharon_Barak	0.183	gcld1, gcld3	0.183	gcld1, gcld3
William_H._Gates_sr	0.163	gcld1,gcld3	0.163	gcld1, gcld3
Douglas_Carswell	0.162	gcld1, gcld3	0.162	gcld1, gcld3
Charlie_Savage	0.158	gcld1, gcld3	0.158	gcld1, gcld3
Lisa_Bloom	0.153	gcld1, gcld3	0.153	gcld1, gcld3
William_Cohen	0.147	gcld_s1, gcld_s2	0.147	gcld_s1, gcld_s2
Boris_Berezovsky_(pianist)	0.143	gcld1, gcld3	0.143	gcld1, gcld3
Lovebug_Starski	0.125	gcld1, gcld3	0.125	gcld1, gcld3
Masaru_Emoto	0.104	gcld_s1,gcld_s2	0.104	gcld_s1, gcld_s2
Rodrigo_Pimentel	0.047	gcld1,gcld3	0.047	gcld1, gcld3
Basic_Element_(music_group)	0.038	gcld1, gcld3	0.038	gcld1, gcld3
Jim_Steyer	0.0	all	0.0	all

There were some differences between the TREC KBA's CCR tasks in 2012 and 2013. In both years, the main task was to filter a large time-ordered stream corpus for documents that are relevant for predefined KB entities. While the main task remained more or less the same in both years, the size and type of the corpus and the predefined list of KB entities either changed and/or were expanded.

3.2.1. Evaluation

We use the official TREC KBA evaluation metrics [5]. Peak F-scores averaged across the entities are used to compare system performances. Also, we use scaled utility (SU), the secondary TREC KBA official metric. SU measures the ability of a system to reject non-relevant documents and accept relevant documents.

3.2.2. Method

We combine all the best parts in the above-mentiomed approaches in an attempt to benefit from the strengths of each. We gathered features from the different approaches and added

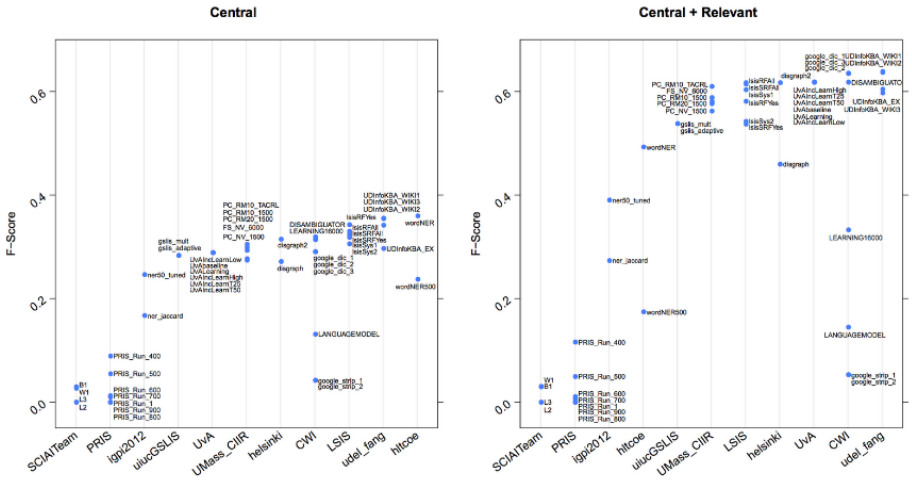


Figure 3.3: Ranked by team's run with highest average F-score (averaged across the 29 target entities).

some new ones making a huge initial feature set. We reduced the feature set using different methods until we had a small powerful subset which we ranked according to information gain. We applied the approach to the 2012 task and our performance was encouraging (both F-measure and SU being above 4.0). Encouraged by our performance on 2012 task, we applied the approach to the 2013 CCR task.

Features

We took 68 features (5 document, 1 entity, 24 document-entity and 38 temporal) from Balog et al. [29, 30]. Document and entity features are computed from processing the documents and entities, respectively. Document-entity features are computed by aggregating scores of strings for which a match has been found in a document. For example, if we consider the Personalized PageRank (PPR) feature, for each entity, there are 100 related entities each with its PPR score. When processing a document entity pair, if a document matches strings from the entity's pre-constructed PPR, we add up the scores and the sum becomes the PPR score for that document-entity pair. We take the 68 features as provided by the authors[1] [29, 30] and add others from [27, 28], described below and some of them modified to suit our approach.

Google's Cross Lingual Dictionary (GCLD)

This is a mapping of strings to Wikipedia concepts and vice versa [26]. The GCLD corpus computes two probabilities: (1) the probability with which a string is used as anchor text to a Wikipedia entity and (2) the probability that indicates the strength of co-reference of an anchor with respect to other anchors to a given Wikipedia entity. We use the product of both for each string.

[1]http://krisztianbalog.com/files/resources/oair2013-kba/runs.zip, accessed in February 2013

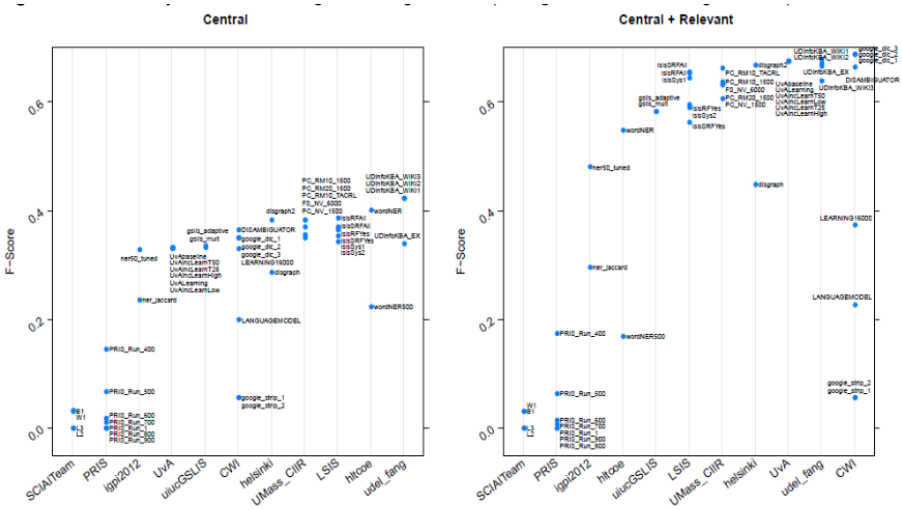


Figure 3.4: Ranked by highest F-score computed from macro-averaged precision and recall.

PPR

For each entity, we computed a PPR score from a Wikipedia snapshot and we kept the top 100 most related entities along with the corresponding scores.

Surface Form (sForm)

For each Wikipedia entity, we gathered DBpedia name variants. These are redirects, labels and names.

Context (contxL, contxR)

From [31], we collected all left and right contexts (2 sentences left and 2 sentences right) and generated n-grams between uni-grams and quadro-grams for each left and right context. Finally, we select the 5 most frequent n-grams for each context.

3.2.3. Training

We use a 2-step classification approach, following [29] and [30], which consists of filtering followed by classification or learning to rank. The first step filters the stream for documents that are potentially relevant using DBpedia name variants of the Wikipedia entities. The second step trains a classification or a learning to rank (LTR) algorithm. In both cases, we treat *central* as positive, and *garbage* and *neutral* as negative examples. However, *relevant* is excluded from the training stage, as these may introduce confusing examples for the classifiers.

For classification, we train the J48 (CL-J48) and the Random forest Model (CL-RF), as implemented in Weka[2]. This mimics the setup of [29].

[2]http://www.cs.waikato.ac.nz/~ml/weka/, accessed in February 2013

Table 3.4: KBA Track 2012. Performances comparison of our approach (lower half) with baselines (upper half).

Method	F-score	SU
MC-RF	0.360	0.263
LTR-RF	**0.390**	**0.369**
CL-RF	**0.402**	**0.396**
LTR-RF	0.394	0.411
CL-J48	0.388	0.306

3.2.4. Result and Analysis

Feature analysis

Our feature selection was done using the 2012 datasets and relevance judgments.

Starting with a smaller feature set, we experimented adding features. We observed that the performances of the three algorithms increase with the addition of features to the initial feature set, reaching a maxima and then decreasing. The increase and decrease are not uniform. However, we observed that the three algorithms reach their respective maxima within the first 13 features. We selected the best F-scores, and they are shown along with three baselines in Table 3.4. We have included two of the highly performing methods on 2012 CCR task as baselines. From classification, the 2-step approach's Random Forest is used as a baseline (MC-RF). The second is LTR's Random Forest (LTR-RF).

The scores in Table 3.4 show that our reduced feature set performs better than the baselines on both performance measures. The most informative features, as measured by information gain and contribution to performance, are: name variants (GCLD), similarity features (cos, jac, kl), related entities (PPR), context, position of entity mention in the document, and length of body text. These features can serve as baseline features for the CCR task.

Results on TREC KBA 2013's CCR

Encouraged by the results on TREC KBA 2012's CCR task, we applied our reduced feature set and the two classification algorithms (J48 and Random Forest) to the 2013 CCR task. We used three sets of features, ranked on the basis of information gain: all 26 features, features up-to FirstPosNorm (FPN) (12 features in total) and Features up-to MentionsBody (MB) (13 features in total). We constructed three training datasets: 2012, 2013, and the union of 2012 and 2013 relevalance judgments, referred to as '1213'. We generated 3 Cl-J48 runs using all features and the training sets, and 9 CL-RF runs using 3 feature sets and 3 training sets.

Some of the entities are grouped, and results are provided per group. Results for the groups for which performance was above the median are shown in Table 3.5. System are named by following the template algorithm_feature set_training dataset_year. For example CL-RF_all_13_13 represents a system using Random Forest with all features, trained on 2013 training dataset with relevance judgments and applied on 2013 CCR task.

Table 3.5 shows our best performance according to micro average F-score and SU. The scores are obtained from the classification confidence scores. We map the scores of irrelevant document-entity pairs to (0, 500] and the scores of relevant to (500, 1000]. For vital

Table 3.5: KBA Track. Our system performances on vital (upper half) and vital+useful (lower half).

System_id	F	SU	Group
CL-RF _all_13_13	0.575	0.571	turing
CL-RF _MB_13_13	0.388	0.404	fargo
CL-RF _all_1213_13	0.353	0.370	hep
CL-RF _FPN_1213_13	0.338	0.333	ocala
CL-RF _FPN_12_13	0.298	0.395	bronfman
CL-RF _MB_13_13	0.290	0.333	wikipedia
CL-RF _FPN_13_13	0.279	0.422	danville
CL-RF _MB_13_13	0.247	0.333	all-entities
CL-RF _MB_13_13	0.241	0.341	hoboken
CL-J48_13_13	0.232	0.333	screenwriters
CL-RF _all_13_13	0.649	0.647	wikipedia
CL-RF _MB_1213_13	0.603	0.602	all-entities

classification, the highest score is on the Turing group. On all entities, the micro-average-F is 0.247, and on Wikipedia entities, it is 0.290. On vital+useful, we did well, achieving a performance of 0.603 on all entities and 0.649 on Wikipedia only.

Our approach was very weak in Twitter entities achieving an F-measure of 0.0. The low performance on Twitter entities is expected since almost all the strong features we used did not apply to Twitter entities. For example, all the similarity (cos, kl, jac), GCLd, PPR, sform and context features were assigned 0 score. We also performed very poorly on the entities groups of startups, french, mining and comedians.

From algorithms, CL-RF performs better in almost all cases. Regarding training dataset, we see that 2013 relevance judgments help train a better model. In many cases, training on 2012 data achieved 0.0 or very low performance. This is probably due to the fact that the CCR task has been changed from its 2012 definition.

Our performance on 2012 CCR task did not translate well to the 2013 CCR task. We suspect that this has to do with the change of the CCR task. However, we have achieved good results for some groups.

3.2.5. Conclusion

We have experimented with string-matching approaches in TREC KBA 2012's CCR task and with machine learning approaches in TREC KBA 2013's CCR task. In the 2012 string-matching approaches, entities are represented by strings obtained from a rich resource called Google Cross Lingual Dictionary (GCLD). In the 2013 TREC KBA, we attempted to combine the best features and approaches from TREC KBA 2012.

Under the GCLD, we have tried different entity representations, different scoring functions, and different scaling functions. We have targeted the central+relevant category because the nature of the strings is not in a position to differentiate between central and relevant. The best score was obtained with no stripping and full per-entity normalization.

Our experiments and approaches show that there are two factors that affect string-matching approached to CCR: entity representation and scoring. A very important point about entity representation is that the entity representations should be used as they are i.e. without lowercasing and stripping off punctuation. The GCLD is noisy, but it also includes many DBpedia labels. The importance of scoring is shown by the results for different scoring functions.

While the GCLD probabilities show how likely a string can be used as an anchor in a link to a Wikipedia entity, they never show how important the anchor text is for a document. The only relationship between the strings in the dictionary and the strings in the document is through string matching. This means a word may have a high probability of being used in a link to a Wikipedia entity, but if the word is not important for the document, the match becomes useless. We believe incorporating some third function that measures the importance of a term for a document can improve the performance. Another challenge is the presence of noise in the GCLD strings (such as "here").

Our machine learning approach in TREC KBA 2012's CCR was different for different entities. On vital+useful, we did well on average on all entities and better on Wikipedia entities. It was very weak in Twitter entities and entities groups of startups, french, mining and comedians, groups that had not much information about them on the Internet. The low performance on Twitter entities and some entity groups is expected.

From machine learning algorithms, CL-RF performs better in almost all cases. Regarding training dataset, we see that 2013 relevance judgments help train a better model. In many cases, training on 2012 dataset and running it on 2013 dataset achieved very low performance. This is probably due to the fact that the CCR task has been changed from its 2012 definition.

Our performance on 2012 CCR task did not translate well to the 2013 CCR task. We suspect that this has to do with the change of the CCR task. However, we have achieved relatively above average results for entities and entity groups for which there was abundant information on the Internet, especially on Wikipedia.

4

Cumulative Citation Recommendation: A Feature-Aware Comparison of Approaches

The work here is an extension of the machine learning approaches for the CCR task discussed in Section 3.2 of Chapter 3. In this chapter, we conduct a feature-aware comparison of approaches to the CCR, a task that aims to filter and rank a stream of documents according to their relevance to entities in a knowledge base. We conducted experiments starting with a big feature set, identified a powerful subset and applied it to comparing classification and learning-to-rank algorithms. With a few powerful features, we achieve better performance than the state-of-the-art. Surprisingly, our findings challenge the previously known preference of learning-to-rank over classification: in our study, the CCR performance of the classification approach outperforms that using learning-to-rank. This indicates that comparing two approaches is problematic due to the interplay between the approaches themselves and the feature sets one chooses to use.

4.1. Introduction

Knowledge Bases such as Wikipedia have gained popularity and can be considered an important knowledge resource in our daily lives. KB curators need to constantly watch for new information and populate and maintain KBs so that they stay up-to-date, useful and accurate. However, the number of entities in a KB on one hand, and the huge amount of new information content on the Web on the other hand makes population and maintenance a challenging task. To address this, the Text REtrieval Conferences (TREC) introduced the KBA track in 2012[1]. TREC KBA seeks to partially automate KB population and maintenance by

[1]http://trec-kba.org/, accessed in July 2015

recommending relevant documents to KB curators. TREC KBA's main task, CCR, aims at filtering a stream to identify those documents that are citation-worthy to KB entities of interest.

A number of studies [27–30] experimented with various types of features and approaches. These studies, while experimenting with a number of features, never examined the power of individual features. Feeding many features into a classifier may, however, make the model unnecessarily complex, increase the chance of overfitting and amplify the curse of dimensionality. Different approaches are, in the absence of common features, compared with each other to determine which one performs better. It is difficult to judge whether the observed performance difference is due to the approaches themselves or the (different) sets of features used.

In this chapter, we study the contributions to performance of individual features with the goal of selecting a few powerful ones. Keeping the set of fixed selected features, we compare the best performing approaches used in the literature. The contributions of the study are: (1) a fair comparison of feature effectiveness from several previous studies, (2) identifying a powerful subset of features leading to an effectiveness beyond that of the state-of-the-art CCR systems, and (3) demonstrating that with the reduced but more effective set of features, previous findings that certain approaches outperform others do not hold, suggesting that we cannot compare approaches independently of the features used.

The rest of the chapter is organized as follows: in sections 4.2, 4.3 and 4.4, we discuss data and problem description, related work, and methods used. In section 4.5, we discuss our experiments, followed by results and analysis in 4.6. Finally, in Section 4.7, we state our conclusions.

4.2. Data and Task description

4.2.1. Data

We use the TREC KBA-CCR-2012 dataset[2] [5]. It consists of 29 Wikipedia entities and a time-stamped stream of documents containing news, social media content, and content from bitly.com URLs.

TREC KBA provided relevance judgments for training and testing. Documents with citation-worthy content to a given entity are annotated as *central*, and those with tangentially relevant content are annotated as *relevant*. Documents with no relevant content and spam are labeled *neutral* and *garbage*.

4.2.2. Task

Given a stream of documents of news items, blogs and social media on one hand and Wikipedia entities on the other, we conduct a feature study to identify a small set of effective features that are then used to compare different approaches employed in CCR.

4.3. Related Work

Three different categories of approaches to solving the task of CCR have been proposed in previous work, categorized as string-matching, classification and learning to rank (LTR).

[2]http://trec-kba.org/kba-ccr-2012.shtml, accessed in July 2015

With string-matching, entities are represented by a small set of key strings that capture entity occurrences, and documents that match the strings are retrieved as relevant [27, 28]. The best performing method uses an entity's name mention and mentions of related entities [28] as features. The method ranks documents by using a function that assigns a base score to a document that mentions the entity by name. Mentions of related entities increase the base score.

The best performing method from the second category compares two multi-step methods. An initial step filters the stream for potentially relevant documents. The 3-step approach uses a classifier to separate *garbage* and *neutral* from *relevant* and *central*, and a second classifier to separate *relevant* from *central*. The 2-step approach directly trains a classifier to separate *garbage* and *neutral* from *central*. Relevant annotations are excluded from the training stage in order not to introduce confusing examples. The 2-step approach achieves a better performance than that of the 3-step approach.

Related to [29], the authors of [30] have proposed to use LTR instead of classification. The classification and learning-to-rank approaches of [29, 30] shared the same set of 68 distinct features. The authors conclude from their experiments that LTR approaches outperform classification approaches.

Our study is an independent reproduction of previously published findings, along with improvements. Specifically, we use the 2-step approach of [29, 30], reconsider the features proposed in [27–30], and demonstrate empirically that a small subset is sufficient and leads to improved results. We demonstrated that with the reduced but more effective set of features, a classification-based approach outperforms a learning-to-rank-based approach. This finding deviates from results in previous study [30].

4.4. Method

We take the 68 features as provided as accompanying data for [29, 30][3] and add 5 others (adapted from [27, 28]), making a total of 73 initial features. The features consist of 5 document, 1 entity, 24 document-entity, 38 temporal, and 5 adapted or new features. Document and entity features are computed from processing the documents and entities respectively. Document-entity features are computed by aggregating scores over strings for which a match has been found in a document. For example, if we consider the Personalized PageRank (PPR) feature, for each entity, there are 100 related entities each with their own PPR score. When processing a document entity pair, if a document matches strings from the entity's pre-constructed related entities, we aggregate the scores and the sum becomes the PPR score for that document-entity pair. Temporal features are meant to capture when important events related to the entities happen by measuring spikes in their respective Wikipedia views and the streaming documents.

The 2-step approach that we use consists of filtering followed by classification (as in [29]) or learning-to-rank(as in [30]). The first step filters the stream for documents that are potentially relevant using DBpedia name variants of the Wikipedia entities. The second step trains a classification or a learning to rank (LTR) algorithm. In both cases, we treat *central* as positive, and *garbage* and *neutral* as negative examples. However, *relevant* is excluded from the training stage not to introduce confusing examples.

[3] http://krisztianbalog.com/files/resources/oair2013-kba/runs.zip., accessed in February 2014

For classification, we use CL-J48 and CL-RF, as implemented in WEKA[4]. For LTR, we use the Random Forest (LTR-RF) approach as implemented in RankLib.[5] Thus, we take the same settings as described in [30] and [29].

4.5. Experiments

4.5.1. Feature reduction

We followed two steps to select a small set of effective features: preliminary elimination and subsequent forward selection. Preliminary elimination was done in two ways. First, we ran an experiment with and without temporal features and observed that the collective contribution of temporal features to performance was negligible. Next, from document-entity features, we excluded all features that are based on partial matching such as features that use the matching of a person's last name. These features are already integrated in our new or adapted features. The preliminary elimination step helps reduce the large feature set to a smaller manageable set for the subsequent forward selection method. After preliminary elimination, there remain 26 features (15 document-entity, 6 document, and 5 new or adapted) listed in 4.1. Next, we apply the forward selection method on these remaining features: add one feature at a time and study its contribution to performance. Based on this, we select an even fewer, but effective set of features.

4.5.2. Baseline runs

We use three baselines, one from each category (string-matching, classification and LTR) that achieves the highest performance. For string-matching, we use [28] (LRE-KBA). For classification, the 2-step approach is used as a baseline (MC-RF). The third baseline, representing the state-of-the-art LTR category, which also uses a 2-step approach, but trains a LTR algorithm instead of a classifier [30] (MC-LTR-RF).

4.6. Result and Analysis

Figure 4.1 shows the performance (F-score) of the three algorithms against feature addition. The features are sorted from left to right, in descending order, in terms of information gain. The plus sign on a feature indicates that we incrementally add the feature into the feature set to the left of it.

From Figure 4.1, we see that the performance of the three algorithms increases with the addition of features to the initial feature set, reaches a maxima and then decreases. We can see that the three algorithms reach their respective maxima within the first 13 features. The addition of features does not improve results (in fact, performance deteriorates). Table 4.2 lists the best F-scores as well as SU for each of the settings.

The results in Table 4.2 show that our reduced feature set performs better than the baselines, on both performance measures. The advantage of having a small set of powerful features is that they are easy to implement. The most informative features, as measured by information gain and contribution to performance, are the name variants (GCLD), similarity features (cos, jac, kl), related entities (PPR), context, position of entity mention in the doc-

[4]http://www.cs.waikato.ac.nz/~ml/weka/, accessed in February 2014
[5]http://people.cs.umass.edu/~vdang/ranklib.html, accessed in February 2014

Table 4.1: Our feature set. The context features are new in the sense they were not used for CCR before. GCLD is as used in [27], and PPR is an adaptation from [28]. The rest of the features are as implemented in [30] and [29].

Feature	Description
GCLD	A mapping of strings to Wikipedia concepts and vice versa [26].
PPR	For each entity, we computed a PPR score from a Wikipedia snapshot, keeping the top 100 entities along with the corresponding scores.
Surface Form (sForm)	For each entity, we gathered DBpedia redirects, labels and names.
Context (contxL, contxR)	From the WikiLink corpus [31], we collected context sentences (2 left and 2 right) and generated n-grams between uni-grams and quadro-grams. We select the 5 most frequent n-grams for each context.
LengthTitle	Term count of document title.
LengthBody	Term count of document body.
LengthAnchor	Term count of document anchor(s).
Source	Document source (news, social, or linking).
English 0,1	Document's language is English or not.
MentionsTitle	Number of occurrences of the target entity in the document title.
MentionsBody	Number of occurrences of the target entity in the document body.
MentionsAnchor	Number of occurrences of the target entity in the document anchor(s).
FirstPos	Term position of the first occurrence of the target entity in the document body.
LastPos	Term position of the last occurrence of the target entity in the document body.
Spread	Spread, i.e., distance between first and last occurrences.
SpreadNorm	Spread, normalized by the document length.
FirstPosNorm	Term position of the first occurrence of the target entity in the document body normalized by the document length.
LastPosNorm	Term position of the last occurrence of the target entity in the document body normalized by the document length.
SpreadNorm	Spread, normalized by the document length.
RelatedTitle	Number of different related entities mentioned in the document title.
RelatedBody	Number of different related entities mentioned in the document body.
RelatedAnchor	Number of different related entities mentioned in the document anchor(s).
jac	Jaccard similarity between the document and the entity's Wikipedia page.
cos	Cosine similarity between the document and the entity's Wikipedia page.
kl	KL-divergence between the document and the entity's Wikipedia page.

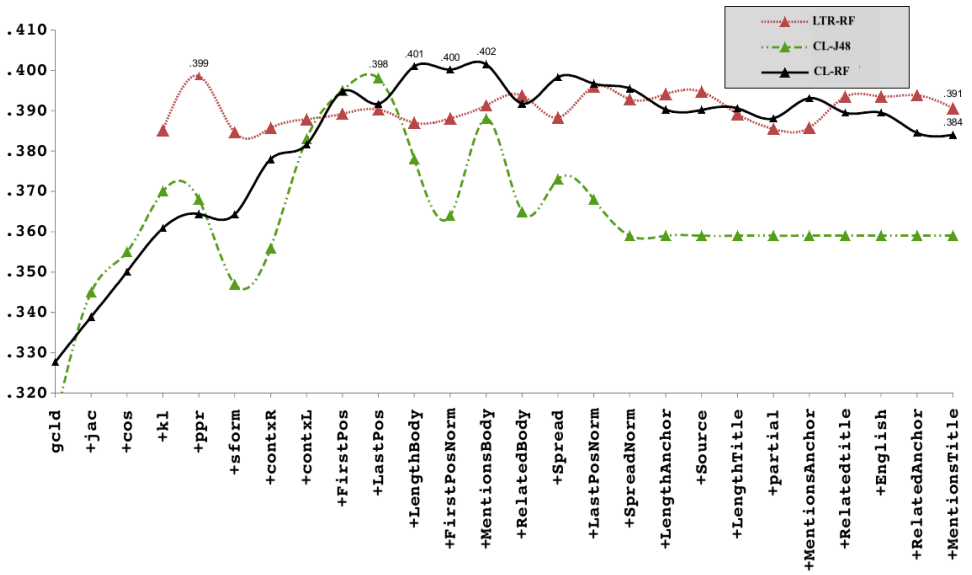


Figure 4.1: Performance (F-score) of classification and LTR algorithms against feature addition. Features are sorted in descending order according to information gain scores.

ument, and length of body text. These features can serve as baseline features for the CCR task.

A surprising observation is that the approach using Classification Random Forest outperforms that using LTR Random Forest. This contrasts with the finding reported in previous work [30], that LTR algorithms outperform classification algorithms. Clearly, the conclusion that a certain approach outperforms another given a set of features does not mean that if the set of features is changed this conclusion still holds.

Random Forest (CL-RF) has achieved the best scores. Since CL-RF results can vary from run to run, it becomes important to check their stability. To do so, we estimated the 95 confidence interval. For each addition of a new feature, we run CL-RF with 10 different random seed initialization and compute the confidence interval. The plot CL-RF in Figure 4.1 is based on the mean performance for 10 different random initializations. The best result achieved with classification Random Forest is 0.402 ± 0.016 (95% confidence limits).

4.7. Conclusion

In this chapter, we have studied the CCR challenge with a focus on feature selection and subsequent comparisons of approaches. We started with a large feature set proposed in the literature, employed a preliminary feature elimination and a subsequent forward selection method to study the contribution of each element of the reduced feature set to performance. We found that with a reduced feature set, improved performance can be achieved compared to the full feature set both in terms of classification and learning-to-rank. We believe having a small selection of powerful features is advantageous because they (1) are easy to implement, and (2) achieve better performance. An important finding is that with the reduced

Table 4.2: Performance comparison of our approach (lower half) with baselines (upper half). Best scores are highlighted.

Method	F-score	SU
MC-RF	.360	.263
MC-LTR-RF	**.390**	**.369**
LRE-KBA	.377	.329
CL-RF	**.402**	**.396**
LTR-RF	.394	.411
CL-J48	.388	.306

but more effective set of features, a classification-based approach outperforms a learning-to-rank-based approach, contradictory to what was found in a previous study [30]. This suggests that when comparing CCR approaches, e.g., classification vs. learning to rank, conclusions do not only depend on the type of classifier or ranker, but also the set of features used, and we should be careful in generalizing conclusions

5

Entity-Centric Stream Filtering and Ranking

CCR is defined as: given a stream of documents on one hand and KB entities on the other, filter, rank and recommend citation-worthy documents. The pipeline encountered in systems that approach this problem involves four stages: filtering, classification, ranking (or scoring), and evaluation. Filtering is only an initial step that reduces the web-scale corpus into a working set of documents more manageable for the subsequent stages. Nevertheless, this step has a large impact on the recall that can be attained maximally. This study analyzes in-depth the main factors that affect recall in the filtering stage. We investigate the impact of choices for corpus cleansing, entity profile construction, entity type, document type, and relevance grade. Because failing on recall in this first step of the pipeline cannot be repaired later on, we identify and characterize the citation-worthy documents that do not pass the filtering stage by examining their contents.

5.1. Introduction

The maintenance of KBs has increasingly become quite a challenge for their curators, considering both the growth of the number of entities considered and the huge amount of online information that appears every day. In this context, researchers have started to create information systems that support the task of CCR: given a stream of documents and a set of entities from a KB, filter, rank and recommend those documents that curators would consider "citation-worthy".

KB curators will expect the input stream to cover all the (online) information sources that could contain new information about the entities in the KB, varying from mainstream news sources to forums and blogs. State-of-the-art CCR systems need to operate on web-scale information resources. Current systems therefore divide up their overall approach in multiple stages, e.g., filtering, classification, ranking (or scoring), and evaluation. This chapter zooms into this first stage, filtering, an initial step that reduces the web-scale input stream into a working set of documents that is more manageable for the subsequent stages. Never-

theless, the decisions taken in this stage of the pipeline are critical for recall, and therefore impact the overall performance. The goal of our research is to increase our understanding how design decisions in the filtering stage affect the citation recommendation process.

We build on the resources created in the KBA track of the Text REtrieval Conference (TREC), introduced in 2012 with Cumulative Citation Recommendation as the main task. As pointed out in the 2013 track's overview paper [10] and confirmed by our analysis of participants' reports, the approaches of the thirteen participating teams all suffered from a lack of recall. Could this be an effect of short-comings in the initial filtering stage?

While all TREC KBA participants applied some form of filtering to produce a smaller working set for their subsequent experiments, the approaches taken vary widely; participants rely on different techniques and resources to represent entities, algorithms may behave differently for the different document types considered in the heterogeneous input stream, and teams use different versions of the corpus. Given these many factors at play, the task of drawing generically applicable conclusions by just comparing overall results of the evaluation campaign seems infeasible. Our work therefore investigates systematically the impact of choices made in the filtering stage on the overall system performance, varying the methods applied for filtering while fixing the other stages of the pipeline.

The main contributions of this chapter are an in-depth analysis of the factors that affect entity-based stream filtering, identifying optimal entity profiles without compromising precision, shedding light on the roles of document types, entity types and relevance grades. We also present a failure analysis, classifying the citation-worthy documents that are not amenable to filtering using the techniques investigated.

The remaining part of the chapter is organized as follows. After a brief related work, Section 5.3 describes the dataset and approach, followed by experiments in Section 5.4. Sections 5.5 and 5.6 discuss the results and a failure analysis. Section 5.7 summarizes our conclusions.

5.2. Related Work

Automatic systems to assist KB curators can be seen as a variation of information filtering systems that "sift through a stream of incoming information to find documents relevant to a set of user needs represented by profiles" [32]. In entity-centric stream filtering, user needs correspond to the KB entities to be curated. However, since the purpose of the filtering component in cumulative citation recommendation is to reduce the web-scale stream into a subset as input for further processing, the decision which documents should be considered citation-worthy is left to later stages in the pipeline.

Other related work addresses the topic of entity-linking, where the goal is to identify entity mentions in online resources and link these to their corresponding KB profiles. Relevant studies include [33, 34], and evaluation resources are developed at the Knowledge Base Population (KBP) track of the Text Analysis Conference (TAC) [35]. Though related, entity linking emphasizes the problem of locating an entity's mentions in unstructured text, where the primary goal of CCR is to identify an entity's most relevant documents.

Our study is rooted in the research carried out in the context of TREC KBA. The problem setup has been essentially the same for both the 2012 and 2013 KBA tracks, but the large size of the 2013 corpus had the effect that all participants resorted to reducing the data-set using an initial filtering stage. Approaches varied significantly in the way they construct

entity profiles. Many participants relied on name variants taken from DBpedia, such as labels, names, redirects, birth names, aliases, nicknames, same-as and alternative names [36–38]. Two teams considered (Wikipedia) anchor text and the bold-faced words of the first paragraph of the entity's Wikipedia page [39, 40]. One participant used a Boolean *and* expression built from the tokens of canonical names [41].

Due to the large variety in the methods applied in different stages of the pipeline, it is difficult to infer which approaches are really the best. By focusing on a single component of the pipeline and analyzing the effects of its design choices in detail, we aim at more generally applicable results.

5.3. Approach

We use the TREC KBA 2013 dataset[1] to compare the effectiveness of different choices for document and entity representation in the filtering stage. Cleansing refers to pre-processing noisy web text into a canonical "clean" text format. In the specific case of TREC KBA, the organizers provide two versions of the corpus: one that is already cleansed, and one that is the raw data as originally collected by the organizers. Entity profiling refers to creating a representation of the entity based on which the stream of documents is filtered, usually by straightforward matching of their textual contents.

5.3.1. Dataset Description

The TREC KBA 2013 dataset consists of three main parts: a time-stamped stream corpus, a set of KB entities to be curated, and a set of relevance judgments. The stream corpus comes in two versions: raw and cleansed. The raw data is a dump of HTML pages. The cleansed version is the raw data after its HTML tags have been stripped off, considering only the documents identified as English (by the Chromium Compact Language Detector[2]). The stream corpus is organized in hourly folders, each of which contains many "chunk files". Each chunk file contains hundreds to hundreds of thousands of semi-structured documents, serialized as thrift objects (one thrift object corresponding to one document). Documents are blog articles, news articles, or social media posts (including tweets). The stream corpus has been derived from three main sources: TREC KBA 2012[3](blogs, news, and urls that were shortened at `bitly.com`), arXiv[4] (e-prints), and spinn3r[5] (blogs).

The KB entities in the dataset consist of 20 Twitter and 121 Wikipedia entities. The entities selected by the organizers of the TREC KBA evaluation are "sparse" (on purpose): they occur in relatively few documents and have an underdeveloped KB entry.

TREC KBA provides relevance judgments, which are given as document-entity pairs. Documents with citation-worthy content to a given entity are annotated as *vital*, while documents with tangentially relevant content, lacking freshliness or with content that can be useful only for initial KB-dossier creation are annotated as *relevant*. Documents with no

[1] `http://trec-kba.org/trec-kba-2013.shtml`, accessed in February 2014
[2] `https://code.google.com/p/chromium-compact-language-detector/`, accessed in February 2014
[3] `http://trec-kba.org/kba-stream-corpus-2012.shtml,accessedin2014`, accessed in February 2014
[4] `http://arxiv.org/`
[5] `http://spinn3r.com/`

relevant content are labeled *neutral*, spam documents are labeled as *garbage*. In total, the set of relevance judgments contains 24162 unique vital-relevant document-entity pairs (9521 vital and 17424 relevant).[6] The relevance judgments have been categorized into 8 source categories: 0.98% arXiv, 0.034% classified, 0.34% forum, 5.65% linking, 11.53% mainstream-news, 18.40% news, 12.93% social and 50.2% weblog. We have regrouped these source categories into three groups, "news", "social", and "other", for two reasons. First, mainstream-news and news are very similar, and can only be distinguished by the underlying data collection process; likewise for weblog and social. Second, some sources contain too few judged document-entity pairs to usefully distinguish between these. The majority of vital or relevant annotations are "social" (63.13%) and "news" (30%). The remaining 7% are grouped as "other".

5.3.2. Entity Profiling

The names of the entities that appear as part of the URL are referred to as "canonical names". For example, entity `http://en.wikipedia.org/wiki/Benjamin_Bronfman` has as its canonical name "Benjamin Bronfman". Similarly, the canonical name of the Twitter entity `https://twitter.com/RonFunchesFor` is "RonFunchesFor". For the Wikipedia entities, we derive additional name variants from DBpedia: name, label, birth name, alternative names, redirects, nickname, or alias. For the 20 Twitter entities, we copied the display names manually from their respective Twitter pages. On average, we extract approximately four different name variants for each entity.

For each entity, we create four entity profiles: canonical (cano), canonical partial (cano-part), all name variants combined (all) and their partial names (all-part). Throughout this chapter, we refer to the last two profiles as name-variant and name-variant partial, using the terms in parentheses in the Table captions.

5.3.3. Evaluation Measures

Our main measure of interest is the recall, as documents missed in this stage cannot be recovered during further processing. We also report the overall performance of a standard high-performing setup for the subsequent stages of the pipeline, which we keep constant. Here, we compute the track's standard evaluation metric, max-F, using the scripts provided [10]. max-F corresponds to the maximally attained F-measure over different cutoffs, averaged over all entities. The default setting takes the vital rating if a document-entity pair has both vital and relevant judgments.

5.4. Experiments and Results

5.4.1. Cleansing: Raw or Cleansed

Tables 5.1 and 5.2 show that recall (on retrieving each relevance judgment) is higher in the raw version than in the cleansed one. Recall increases on Wikipedia entities vary from 13% to 16.4%, and on Twitter entities from 62.8% to 358%. At an aggregate level, recall improvement ranges from 15% to 20.5%. The recall increases are substantial. To put it into perspective, an 15% increase in recall on all entities is a retrieval of 2864 more unique

[6] The numbers of vital and relevant do not add up to 24162 because some documents are judged as both vital and relevant, by different assessors.

Table 5.1: Vital Recall for Cleansed.

	cano	cano-part	all	all-part
Wikipedia	61.8	74.8	71.5	77.9
Twitter	1.9	1.9	41.7	80.4
Aggregate	51.0	61.7	66.2	78.4

Table 5.2: Vital Recall for Raw.

	cano	cano-part	all	all-part
Wikipedia	70.0	86.1	82.4	90.7
Twitter	8.7	8.7	67.9	88.2
Aggregate	59.0	72.2	79.8	90.2

document-entity pairs.

5.4.2. Entity Profiles

The aggregate recall increase from canonical partial to name-variant partial is 25% and from canonical names to name variants is 35% (see Table 5.2). This means that a quarter of the documents mentioned the entities by partial names of non-canonical name variants and more than one-third of the documents mention the entities by non-canonical names, respectively. Generally, recall increases as we move from canonical to canonical partial, to name-variant, and to name-variant partial. The only exception is that using canonical partial leads to a better recall for Wikipedia entities than using the name-variants.

5.4.3. Relevance Rating: Vital and Relevant

The primary objective of cumulative citation recommendation is to identify the citation-worthy documents. We would like to know if there is a difference between filtering vital and relevant documents (as measured by recall). This could be helpful to make choices that improve the retrieval of citation-worthy documents selectively. In Table 5.3, we observe that recall performances considering vital documents only are in general higher than those that consider relevant documents as well. Especially for Wikipedia entities, the vital documents tend to mention the entities by their canonical name. This observation can be explained by the intuition that a highly relevant document will usually mention the entity multiple times, using different forms to refer to it. Those documents are therefore likely to pass the filtering stage.

5.4.4. Document Categories and Entity Types

The study of recall across document categories (news, social, other) helps us understand how types of documents behave with respect to filtering. Our documents are divided mainly between social and news. Table 5.3 shows that for Wikipedia entities recall for news docu-

Table 5.3: Breakdown of recall performances by document source category. Vit is for Vital, Rel is for Relevant, cano-p is for cano part, and all-p for all part.

		Aggregate			Wikipedia			Twitter		
		other	news	social	other	news	social	other	news	social
Vit	cano	82.2	65.6	70.9	90.9	80.1	76.8	8.1	6.3	30.5
	cano-p	90.4	80.6	83.1	100.0	98.7	90.9	8.1	6.3	30.5
	all	94.8	85.4	83.1	96.4	95.9	85.2	81.1	42.2	68.8
	all-p	100	99.2	95.9	100.0	99.2	96.0	100	99.3	94.9
Rel	cano	84.2	53.4	55.6	88.4	75.6	63.2	10.6	2.2	6.0
	cano-p	94.7	68.5	67.8	99.6	97.3	77.3	10.6	2.2	6.0
	all	95.8	90.1	72.9	97.6	95.1	73.1	65.2	78.4	72.0
	all-p	98.8	95.5	83.7	99.7	98.0	84.1	83.3	89.7	81.0
All	cano	81.1	56.5	58.2	87.7	76.4	65.7	9.8	3.6	13.5
	cano-p	92.0	72.0	70.6	99.6	97.7	80.1	9.8	3.6	13.5
	all	94.8	87.1	75.2	96.8	95.3	75.8	73.5	65.4	71.1
	all-p	99.2	96.8	86.6	99.8	98.4	86.8	92.4	92.7	84.9

ments is higher than for social. In Twitter entities, however, the recall for social documents is higher than for news, except in name-variant partial. Regarding the two types of entities (Wikipedia and Twitter), we see that Wikipedia entities achieve higher recall than Twitter entities (see Tables 5.1, 5.2 and 5.3).

5.4.5. Impact on Classification

We now will conduct experiments to see how the different choices we made at the filtering stage impact the subsequent steps of the pipeline. Based on the findings of previous work [29, 30, 42], we use a standard pipeline, where the documents passing the filtering stage are classified into their relevance grades. We take the then state-of-the-art WEKA's[7] Classification Random Forest and the set of features used in [42], and the resulting classifier is known to be effective for the CCR problem. We follow the official TREC KBA training and testing setting, that is, we train on the number of documents that our filtering system retrieves from the training data and test on those documents retrieved from the test set. For example, when we use cleansed data and canonical profile, we train on training relevance judgments that we retrieve from the cleansed corpus, using the canonical profile, and test on the corresponding test relevance judgments that we retrieve from the cleansed corpus. The same applies to other combinations of choices. In here, we present results showing how the cleansing, entity type, document category, and entity profile impact classification performance.

Tables 5.4 and 5.5 show the max-F performance for vital relevance ranking. On Wikipedia entities, except for canonical entity profiles, the max-F performance using the cleansed version of the corpus is better than that using the raw one. On Twitter entities however, the performance obtained using the raw corpus is better on all entity profiles, except for name-variant partial. This result is interesting, because we saw in previous sections that *recall*

[7] `http://www.cs.waikato.ac.nz/~ml/weka/`, accessed in July 2015

Table 5.4: Cleansed: vital max-F.

	cano	cano-part	all	all-part
All-entities	0.241	0.261	0.259	0.265
Wikipedia	0.252	0.274	0.265	0.271
Twitter	0.105	0.105	0.218	0.228

Table 5.5: Raw: vital max-F.

	cano	cano-part	all	all-part
all-entities	0.240	0.272	0.250	0.251
Wikipedia	0.257	0.257	0.257	0.255
twitter	0.188	0.188	0.208	0.231

when using the raw corpus is substantially higher than using cleansed one. This gain in recall for the raw corpus does however not translate into a gain in max-F for recommending vital documents. In fact, in most cases overall CCR performance decreased. Canonical partial for Wikipedia entities and name-variant partial for Twitter entities achieve the best results. Considering the vital-relevant category (Tables 5.6 and 5.7), the results are different. The raw corpus achieves better results in all cases (except in the canonical partial of Wikipedia). Summarizing, we find that using the raw corpus has more effect on relevant documents and Twitter entities.

5.5. Analysis and Discussion

There are 3 interesting observations. 1) cleansing impacts relevant documents and Twitter entities negatively. This is validated by the observation that recall gains in Twitter entities and the relevant categories in the raw corpus also translate into overall performance gains. Cleansing removes more relevant documents than it does vital, which can be explained by the fact that it removes related links and adverts that may contain a mention of the entities. One example we saw was that cleansing removed an image with the text of an entity name that was actually relevant. Cleansing also removes more social documents than news, as can be seen by the fact that most of the missing documents from cleansed are social documents. Twitter entities are affected because of their relation to relevant documents and social documents. Examination of the relevance judgments shows that about 70% of relevance judgments for Twitter entities are relevant.

2) Taking both performance (recall at filtering and overall F-score) into account, the trade-off between using a richer entity-profile and retrieval of irrelevant documents results in Wikipedia's canonical partial and Twitter's name variant partial as the two best profiles for Wikipedia and Twitter respectively. This is interesting because TREC KBA participants did not consider Wikipedia's canonical partial as a viable entity profile. Experiments with richer profiles for Wikipedia entities increase recall, but not overall performance.

Table 5.6: Cleansed: vital-relevant max-F.

	cano	cano-part	all	all-part
all-entities	0.497	0.560	0.579	0.607
Wikipedia	0.546	0.618	0.599	0.617
twitter	0.142	0.142	0.458	0.542

Table 5.7: Raw: vital-relevant max-F.

	cano	cano-part	all	all-part
all-entities	0.509	0.594	0.590	0.612
Wikipedia	0.550	0.617	0.605	0.618
twitter	0.210	0.210	0.499	0.580

3) The analysis of entity profiles, relevance ratings, and document categories reveals three differences between Wikipedia and Twitter entities. a) Wikipedia entities achieve higher recall and higher overall performance. b) The best profiles for Wikipedia entities are canonical partial and for Twitter entities name-variant partial. c. The fact that Twitter canonical names achieve very low recall means that documents (especially news and others) rarely use Twitter usernames to refer to Twitter entities. However, comparatively speaking, social documents refer to Twitter entities by their usernames rather than news and others suggesting a difference in adherence to standards in names and naming.

The high recall and subsequent higher overall performance of Wikipedia entities can be due to two reasons. First, Wikipedia entities are relatively better described than Twitter entities. The fact that we can retrieve different name variants from DBpedia is an indication of rich description. On the contrary, the fact that Twitter's richest profile achieves both the highest recall and the highest max-F scores indicates that there is still room for enriching the Twitter entity profiles. Rich description plays a role in both filtering and computation of features such as similarity measures in later stages of the pipeline. By contrast, we have only two names for Twitter entities: their usernames and their display names. Second, unfortunately, no standard DBpedia-like resource exists for Twitter entities, from which alternative names can be collected.

In the experimental results, we also observed that recall scores in the vital category are higher than in the relevant category. Based on this result, we can say that the more relevant a document is to an entity, the higher the chance that it will be retrieved with alternative name matching. Across document categories, we observe a pattern in recall of others, followed by news, and then by social. Social documents are the hardest to retrieve, a consequence of the fact that social documents (tweets and blogs) are more likely to point to a resource where the entity is mentioned, mention the entity with short abbreviation, or talk without mentioning the entities but with some context in mind. By contrast news documents mention the entities they talk about using the common name variants more than social documents do. However, the greater difference in percentage recall between the different entity profiles in the news

Table 5.8: The number of documents missing from raw and cleansed extractions (upper part cleansed, lower part raw).

Category	Vital	Relevant	Total
Cleansed	1284	1079	2363
Raw	276	4951	5227
Missing only from cleansed	1065	2016	3081
Missing only from raw	57	160	217
Missing from both	219	1927	2146

category indicates news refers to a given entity with different names, rather than by one standard name.

5.6. Failure Analysis: Vital or Relevant, but Missing

The use of name-variant partial for filtering is an exhaustive attempt to retrieve as many relevant documents as possible, at the cost of bringing in many irrelevant documents. However, we still miss about 2363 (10%) of the vital-relevant documents. If these are not even mentioned by their partial name variants, what type of expressions were they mentioned by?

Table 5.8 shows the documents that we miss with respect to cleansed and raw corpus. The upper part shows the number of documents missing from cleansed and raw versions of the corpus. The lower part of the table shows the intersections and exclusions in each corpus.

One would naturally assume that the set of document-entity pairs retrieved from the cleansed corpus would be a sub-set of those that are retrieved from the raw corpus. We find that this is however not the case; we even find that we retrieve documents from the cleansed corpus that we miss from the raw corpus. Examining the content of the documents reveals that this can be attributed to missing text in the corresponding document representations. Apparently, a (part of) the document content has been lost in the cleansing process, where the removal of HTML tags and non-English content resulted in a loss of partial or entire content. Documents missing from the raw corpus are all social ones (tweets, blogs, posts from other social media), where the conversion to the raw data format (a binary byte array) may have faulted. In both cases, the entity mention happens to be on the part of the text cut out in the transformation.

The most surprising failures correspond to judged documents that do not pass the filtering stage, neither from the raw nor from the cleansed version of the corpus. These may indicate a fundamental shortcoming of filtering the stream using string-matching, requiring potentially more advanced techniques. Our failure analysis identifies 2146 unique document-entity pairs, the majority (86.7%) of which are social documents, 219 of these judged as vital, and related to 35 entities (28 Wikipedia and 7 Twitter).

We observed that among the missing documents, different document identifiers can have

the same content, and be judged multiple times for a given entity.[8] Excluding duplicates, we randomly selected 35 distinct documents, 13 news and 22 social, one for each entity. Based on this subset of the judgments, we categorized situations under which documents can be vital, without mentioning the entity in ways captured by the entity profiling techniques investigated.

1. Outgoing link mentions: posts with outgoing links mentioning the entity.

2. Event place - event: A document that talks about an event is vital to the location entity where it takes place. For example Maha Music Festival takes place in Lewis and Clark_Landing, and a document talking about the festival is vital for the park. There are also cases where an event's address places the event in a park and due to that the document becomes vital to the park.

3. Entity - related entity: A document about an important figure such as an artist, athlete can be vital to another. This is especially true if the two are contending for the same title, one has snatched a title, or award from the other.

4. Organization - main activity: A document that talks about an area in which the company is active is vital for the organization. For example, Atacocha is a mining company and a news item on mining waste was annotated vital.

5. Entity - group: If an entity belongs to a certain group (class), a news item about the group can be vital for the individual members. FrankandOak is named innovative company and a news item that talks about the group of innovative companies is relevant to it.

6. Artist - work: Documents that discuss the work of artists can be relevant to the artists. Such cases include books or films being vital for the book author or the director (actor) of the film. Robocop is a film whose screenplay is by Joshua Zetumer. A blog that talks about the film was judged vital for Joshua Zetumer.

7. Politician - constituency: A major political event in a certain constituency is vital for its politicians. Take e.g. a weblog that talks about two north Dakota counties being drought disasters. The news is considered vital for Joshua Boschee, a politician, who is a member of the North Dakota democratic party.

8. Head - organization: A document that talks about an entity's organization can be vital: Jasper_Schneider is USDA Rural Development, state director for North Dakota, and an article about problems of primary health centers in North Dakota is judged vital for him.

9. World knowledge, missing content, and disagreement: Some judgments require world knowledge. For example "refreshments, treats, gift shop specials, …free and open to the public" is judged relevant to Hjemkomst_Center. Here, the person posting this on social media establishes the relation, not the text itself. Similarly "learn about the

[8] For a more detailed analysis of the effect of duplicate documents on evaluation using the KBA stream corpus, refer to [43].

> gray wolf's hunting and feeding …15 for members, 20 for nonmembers" is judged vital to Red_River_Zoo.

For a small remaining number of documents, the authors found no content or could otherwise not reconstruct why the assessors judged them vital.

5.7. Conclusion

In this chapter, we examined the effect of the chain of interactions of cleansing, entity profiles, the effect of the type of entities (Wikipedia or Twitter), categories of documents (news, social, or others) and the relevance ratings (vital or relevant) on recall and overall performance. There is a difference between vital and relevant rankings with respect to filtering: it is easy to achieve higher recall for vital documents only than for the group of "vital or relevant" ones. Given the importance of vital documents (those are the ones we do not want to miss), this is good news for the development of high performing CCR systems.

Cleansing may remove (partial) document content, thereby reducing recall up to 21%. But, this affects the performance of retrieving the relevant documents more than that of vital ones. Looking beyond recall, the overall performance on ranking vital documents improves for Wikipedia entities. Considering the relevant documents, cleansing affects overall performance negatively. If one is interested in vital documents, then we recommend cleansing, but if one is interested in relevant documents too, then cleansing seems disadvantageous. For KB curation, the emphasis is likely on vital documents, but other tasks (such as filtering information for journalists) may require high performance on both relevance grades.

Regarding entity profiles, the most effective profiles of Wikipedia entities rely on their canonical partial representation, while the partial name variants perform best for Twitter entities. Because entity type and relevance grade both exhibit differences regarding filtering, they should be dealt with differently to maximize performance. Similarly, social posts and news should be treated differently.

Despite an exhaustive attempt to retrieve as many vital documents as possible, we observe that there are still documents that defy retrieval. About 10% of the vital or relevant documents cannot be identified using our entity profiling techniques, establishing a 90% recall as an upper bound for the full pipeline. The circumstances under which this happens are many. We found that some judged documents are not fully represented in the collection, and in a few cases, it is simply not clear why assessors deemed those documents vital. However, the main circumstances under which vital documents can defy filtering can be summarized as an outgoing link mentions, venue-event, entity - related entity, organization - main area of operation, entity - group, artist - artist's work, party - politician, and world knowledge. More advanced entity profiling techniques will be necessary to resolve these situations in the future.

6

New Developments in CCR and Related Tasks

6.1. Introduction

In Part II of the thesis, we have investigated string-matching and machine learning approaches to the task of CCR, which is filtering and ranking a stream of documents according to their citation-worthiness to an KB entity (with a rich profile) in order to accelerate the maintenance and population of knowledge bases. In this chapter, we review and discuss research developments in the area with a focus on those that have impact on our experiments and findings,

TREC KBA ran in 2012, 2013, and 2014. The main task remained the same over the years: given a KB entity with a rich profile, filter and rank a stream of documents according to their citation-worthiness to an entity to accelerate the maintenance and population of knowledge bases. Over the three years, however, the query entities changed in number and type; the datasets grew by subsuming the previous year's dataset. New tasks (for example slot filling) around the main task have been defined. We participated only in 2012 and 2013, and in both years, our participation was only on the main task.

In 2014, the task remained basically the same, but the size of the dataset grew and the slot-filling task was also redefined to make it simpler [44]. Additional metadata on the stream documents were also introduced and more focus was given for long-tail query entities.

In all of the three years, high-scoring participants have used different approaches including feature engineering for query entity representation and document representation, using names of related entities, and various types of classifiers. Our approaches in 2012 and 2013 are also a combination of these approaches. Our 2012 participation focused on the string-matching approach for the CCR task and was among the highest-performing approaches. Our attempt to use the best features from 2012 for our machine learning approach in 2013 was not as expected. This was a reason for us to research the interplay between features and machine learning algorithms.

Chapters 4 and 5 are extensions of our works in the first two chapters. Chapter 4, which is

the extension of our participation in 2013, explores the interplay between feature choices and machine learning algorithms. In the TREC KBA participation, participants' systems were evaluated based on overall performance. Our experiment in 2013, where we tried to combine the best features from 2012 and the then best state-of-the-art algorithms, did not readily translate into success. That prompted us to further investigate the interplay between feature sets and the algorithms in the overall performance. We found that features and algorithms have so strong interplay that comparing two algorithms that use different feature sets is erroneous.

Chapter 5, entity-centric stream filtering, deals with the filtering stage of the CCR task. This was also an extension of our work on TREC KBA 2012 and 2013 since it was the filtering approaches we used in both cases that prompted us to further investigate this stage of the CCR task. In this study, we explored the "upper bound" on recall and conducted an error analysis to further understand the causes of documents not being filtered.

TREC KBA as a benchmarking initiative did not continue running after 2014 and was succeeded by TREC Dynamic Domain Track, which concerns itself with dynamic, exploratory search within information domains [45]. But the interest in the entity-centric stream filtering task continues [14, 46–49]. While there are new approaches and use of new features to the CCR task [46], there have not been new investigations on the interplay between features and the machine learning algorithms. To the best of our knowledge, there are also no new studies focusing on the entity-centric stream filtering task either.

Systems approaching the TREC KBA's CCR task involve pipelines consisting of several different sub-tasks: filtering, entity mention detection, entity-linking, entity disambiguation and entity-centric document ranking. New developments in one or more of these areas impact the CCR task. In the following section, we briefly review developments in neural network approaches and discuss their relation and implication for the CCR task and sub-tasks.

6.2. Neural Networks

Word embeddings, sub-word tokenization and deep neural network approaches have resulted in great improvement in NLP tasks such as machine translation, language modeling, entity disambiguation and entity linking. Word embeddings are techniques for turning words in a corpus to vector representations of the words, which are amenable for computational manipulation. The essence of word embeddings is that words that appear in a similar context must have similar vectors [50]. While word embeddings can be used directly to accomplish different tasks, they are mainly useful as inputs to deep neural network approaches. Word embeddings themselves are usually generated by a neural network technique called word2vec [51]. One can train their own word embeddings, but most of the time, pre-trained word embeddings are used.

A weakness of word embeddings is not being able to efficiently handle out-of-vocabulary (OOV) words and polysemy. To address the OOV weakness, a number of sub-word tokenization approaches have been proposed [52, 53]. One of these sub-word tokenization techniques is SentencePiece [53], an open source, unsupervised tokenization and detokenization approach that allows lossless conversion between the sub-word tokens and the original text. The sub-word tokens are then converted to embeddings and usually concatenated to the word embeddings before they are fed as inputs to a neural network approach. This sub-word tokenization approach helps overcome the OOV weakness because sub-parts of OOV words are

usually parts of regular words. Sub-word tokenization such as SentencePiece replace expensive and language-specific word tokenization and allow for the development of end-to-end neural network approaches for NLP tasks.

The basic architecture of deep neural network approaches for sequence transduction has an encoder into which input is converted, and a decoder that reads from the encoder and produces a sequence of strings. Deep neural network approaches for NLP sequence transduction tasks have also undergone several changes and developments. First feed-forward Recurrent Neural Network (RNN) approaches were most popular with NLP tasks. RRNs, however, fail to capture long-distance dependencies. To address this, Long Short-Term Memory (LSTM) neural networks were used. LSTMs store information in cell states. At every stage, an LSTM takes the previous cell's state and output, and a new input to generate a new cell's state and a new output. LSTMs, however, face the same problems when dealing with long-distance dependencies in sentences.

To address the above limitations of RNNs and LSTMs, attention was proposed [54]. In attention, each word has its own hidden state and all hidden states are passed to the decoder. Then, the hidden states are used at each step of the RNN to decode. The use of RNNs here, however, does not allow parallel processing, which is necessary in dealing with big corpus. Convolutional Neural Network(CNN) can help overcome the problem of parallelization because each word can be processed independently, that is, it does not need the translation of the previous word for it to be translated. But, CNNs do not necessarily help with long-distance dependencies.

Here, Transformers come to the rescue [55]. A transformer has encoders and decoders. All encoders have the same architecture. Each encoder has two layers: a self-attention layer and a feed-forward layer. The decoder also has two layers and one more layer, called encoder-decoder attention, between the two layers. The encoder receives word embeddings (vectors) of the input words. Once they pass this stage, the inputs can be processed in parallel in the feed-forward layer. Attention and transformer enable contextual representation of words making it possible to capture polysemy. They are very popular at the moment in NLP research.

Attention and the transformer architecture have shown great improvements in machine translation [52, 56], language modeling [57, 58], entity disambiguation [59], and entity linking [60–62].

Transfer Learning, which was previously shown to be useful in computer vision, has been applied to NLP. Attention and transformer have been deployed to pre-train Large Language Models (LLMs), and those language models are applied in downstream tasks. Prominent examples are GPT [63], ELMo [57] and BERT [58]. These language models contain contextualized word representations and can be useful for a lot of downstream machine learning tasks, where they are used either as additional features or are fine-tuned with additional domain-specific training.

Language models are found to store different linguistic and semantic information about words, as well as factual and commonsense knowledge. Petroni, et al. [64] examined the state-of-the-art language models for their effectiveness in storing commonsense knowledge in comparison to traditional knowledge bases built by manual or carefully crafted relation extraction techniques. They report that pre-trained language models outperform traditional knowledge bases in several tasks such as open-domain question answering. This is an in-

teresting finding, especially given the fact that the language models are relatively easier to build and maintain, as compared to say knowledge graphs that require extensive work and maintenance [65].

ELMo [66] and BERT [67] have been investigated for their capability to model entities [68]. Their contextualized word representations, from which entity representations can be derived, have shown strong improvement over static word embeddings on several entity-related tasks [68]. While ELMo representations are capable of performing zero-shot tasks [68], BERT-derived entity representations need further supervised training. For example, BERT has been fine-tuned to perform a NER task in this blog[1], using an annotated GMB (Groningen Meaning Bank) corpus for entity classification[2].

Several other studies directly apply deep neural network approaches to entity-related tasks. Some deep neural network approaches to entity-related tasks make use of two types of contextual information: local information based on words that occur in a context window around an entity mention, and global information, based on the whole document [59]. Ganea et al. [59] use entity embeddings to capture entity representation and local context. Entity embeddings are bootstrapped from word embeddings and are further trained, independently, with contexts for each entity. Contexts are obtained from a Wikipedia page and from the context windows on hyperlinks to the Wikipedia page. Instead of using all context words in a window, they focus on only a few relevant context words. Attention-based learned combination of mention prior and context-based entity score produces a final entity-mention local context score. Document-level coherence takes the mention, the local context and the entity embeddings and learns a deep neural network model, followed by a final CRF loopy belief propagation inference.

Kolistas et al. [61], on the other hand, propose an end-to-end neural entity linking approach that is also designed to capture the dependency between the mentioned detection and the entity disambiguation sub-tasks. Text spans mentioning potential candidates are first generated. Each mention-candidate pair gets a context-aware compatibility score based on word and entity embeddings, followed by neural attention and global voting mechanism.

They used Ganea et al.'s [59] entity embeddings and candidate selection. A score for mention-entity pairs is computed as a dot product of embeddings. That score is combined with a log-prior probability using a shallow feed-forward neural network. At training, all spans are identified and mention-entity pairs that are in the gold standard are rewarded and non-linking pairs are penalized. At inference, mention-entity pairs with a score higher than a threshold are then selected. Finally, a global disambiguation is performed by comparing the cosine similarity score of an entity's embeddings and the normalized average of all other voting entities' embeddings.

The above approaches to entity linking deal with entities that are seen in the training set. Logeswaran et al. [69] extend entity linking to unseen entities, as opposed to entities that are observed during training. However, they assume labeled mention-entity pairs for training. They construct training and evaluation datasets for training and evaluation from fandom (formerly Wikia)[3]. The datasets contain labeled mentions, which can be automatically linked to the entity descriptions using, for example, string matching. During evaluation, they take

[1]https://www.depends-on-the-definition.com/named-entity-recognition-with-bert/, accessed in 2022
[2]https://www.kaggle.com/abhinavwalia95/entity-annotated-corpus, accessed in November 2023
[3]https://www.fandom.com/, accessed in November 2023

the labeled mentions only, and generate candidate entities followed by candidate ranking. The candidates are generated using BM25 between the mention text and the entity descriptions. They take top-K candidates, which are fed to the candidate ranking stage. In the ranking, mention context and entity description embeddings (which are WordPiece [52] tokens) are concatenated, forming a pair that can attend to each other and which are used to train a BERT-base model, according to which the candidate entities are ranked. To adapt the model to unseen entities, they insert a domain adaptive pre-training, where the model is trained on the target domain corpus.

A different approach is employed in DeepType [60], where they show a way to automatically build an entity type system, from structured (Wikidata, Wikipedia) and unstructured sources, and use the type system to train a deep neural network, which is then used to perform entity linking. The type system is used to constrain the neural network's output to respect the type system. For example, it forces the system to output either person, organization or place. This approach achieves state-of-the-art results on entity linking on standard datasets. It also, however, needs labeled mention-entity pairs for training and evaluation.

The works reviewed above make use of labeled training and evaluation datasets. Datasets with pairs of entities and mentions such as AIDA/CoNLL [70], AIDA CoNLL-YAGO [71], ACE 2004 [59], and several other similar datasets are used for training and evaluation. Several of the papers make their code and sometimes their datasets available. Attempts to replicate, with minimum effort, some of the experiments, did not succeed. This is in line with a recent study that attempted to replicate and reproduce many top deep neural network approaches to the task of top-N recommendation [72], but found only a few of them were reproducible. We were able to partially replicate the work in this blog:https://www.depends-on-the-definition.com/named-entity-recognition-with-bert/[4]. The blog reports an F1 score of 0.785 and an accuracy score of 0.9879. Our replication of the work did not, however, produce the same performance scores. Our replication's performance was an F1 score of 0.420 and an accuracy score of 0.914. We reran it three times, and it showed some small variations, most probably due to the random split in the training and evaluation sets. Our F1 scores were, however, nowhere close to the blog's reported F1 score. We note, however, that other studies also report F1 performances that are similar to that of the blog's [67].

State-of-the-art entity mention detection and entity linking tools (which accomplish both mention detection and linking to an entity in a knowledge base) are now either based on deep neural networks, leverage LLMs or incorporate either of them. Flair [73] which is used as a state-of-the-art entity-mention detection tool is based on word embeddings. End-to-end entity linker ReFiNED [74] is based on transformers. REL (Radboud Entity Linker) [75] which uses Flair for entity mention detection uses a neural network. GENRE [76] and BLINK [77] use fine-tuned BERT architectures.

6.3. Relation to TREC KBA's CCR Task

Some of the neural network approaches for entity mention detection, entity-linking, or entity disambiguation presented above aim to identify entity mentions and then correctly resolve them to the KB entities they refer to. The CCR task of TREC KBA, on the other hand, is the task of filtering and ranking stream documents according to their relevance (citation-

[4]https://www.depends-on-the-definition.com/named-entity-recognition-with-bert/, accessed in November 2023

worthiness) to a given entity. Training examples are, therefore, at the entity-document level with entity-document pairs being labeled either "central", "relevant", "neutral" or "garbage". It is, therefore, the whole document and not a particular mention, that is being linked to a KB entity. Mentioning spans are not identified in the training set.

Adapting the neural network approaches above to the CCR task needs some serious changes. The first order of business would be to change the entity-document level training set to a mention-entity level. This is, however, difficult for several reasons. One reason is that mention detection is not very easy [78]. Even if one opts for accomplishing mention detection using string matching or other approaches (for example using models trained according to [61, 62, 62]) or some of the state-of-the-art entity linkers, there might be several mentions of an entity in a document. There needs to be some way of combining the mentions, or determining the mention that is the reason for an entity-document pair being labeled either one of the labels of "central", "relevant", "neutral" or "garbage"

Another striking difference between TREC KBA's CCR task and the entity-related tasks above is the size of the datasets they deal with. CCR systems must deal with terabytes of data, calling for approaches involving big data frameworks such as Hadoop. The above deep neural network approaches are meant to work mostly with small training and test data. They may therefore not be directly easily applicable to the CCR task that we tackled in this part of the thesis.

The CCR task can be seen as a pipeline involving many steps. One can think of ways of employing some of the neural network approaches to parts of the CCR pipeline. The filtering stage, which is needed to reduce the big stream corpus to a manageable set of documents, can be seen as a ranking problem where documents from the stream are ranked according to their relevance to an entity from the preselected entities. A cutoff can then be used to select the top N potentially relevant documents. A cruder approach can be to use entity mention detection and then filter those documents that mention any of the preselected entities as potentially relevant. Another way is to first expand the entity in the entity-document pair to its profile (for example its Wikipedia profile) and then view the CCR task as a profile-document transduction task.

By treating profile-document as a sequence, we can train a neural network model using the TREC KBA training set. The profiles and documents can be represented with different document embedding techniques [79, 80] and be used to train a deep neural network model. This, however, has some challenges too. In a document, the relevant information for an entity is usually found in a sentence or a passage. This is why, in the entity linking tasks, a context window around an entity mention is used to capture the relevant context [59, 61]. The rest of the document text may be a noise when we consider the document's relevance to a particular entity.

Another approach might be to use the neural ranking model used in ad-hoc information retrieval [81, 82]. The CCR task can be cast as a text ranking problem where documents are ranked according to their relevance to the preselected entities and deep learning approaches [81] or pre-trained transformers [82] can then be used to accomplish the task. Very relevant to the CCR task are two works that use BERT as a neural ranking model [83, 84]. In these two works, BERT is adapted by training it using domain-specific datasets of query-document sequences. The assumption here is that "text retrieval requires understanding both the text content and the search task" [83]. In [83], they use search logs to train BERT followed by

fine-tuning on a news dataset and general webpage dataset. In [84], they use several datasets for training, of which microblog datasets were the most effective.

The approaches in [83, 84] can be adapted to the CCR task. One approach is, following [83], to train BERT on profile-document sequences from the TREC KBA training data. Since BERT is more suited to sentence-sentence sequences, shorter profiles (for example, the first paragraph of the entity's Wikipedia profile) can be used.

During inference (ranking), document context windows that are potentially citation-worthy or relevant information to an entity can be identified by a string-matching system and then independently scored by the model. The top-K relevant windows can be combined in different ways. One way is to consider only the highest-scoring window, or an average of the top-K context windows [83, 84]. The highest context window can also be combined with the overall document score, as is done in [84].

A more promising approach might be to use neuro-symbolic approaches (combining neural representations, string-matching approaches and knowledge graphs) as discussed in [85, 86].

6.4. Conclusion

To conclude, we see some opportunities to apply neural network approaches to the CCR task. Mentions and contexts can be detected in two ways. One is using the canonical and variant names we used in our approach. A second approach can be to use a neural mention detection model trained according to [61, 62]. Using the detected mentions, we can identify context windows (or sentences) in a document. We can then apply the entity embedding and document-level coherence techniques used in [59, 61] to capture the local context of the mentioned entity and the global coherence of the referenced entity. Alternatively, we can employ the neural ranking model that leverages the BERT language model as is done in [83, 84].

There is a relationship between our string-matching approach using rich entity representations performing well and the success of sub-word tokenization methods (including the compression algorithm Byte-Pair Encoding (BPE) [87]) and neural approaches, indicating that the frequency of mention of (parts of) the query terms in a document is related with the document being relevant to the query. Findings about LLM's ability to memorize factual knowledge about entities also confirm this. Recent studies have shown that the more frequent mentions of an entity there are in input data, the better that LLMs will memorize facts about the entity and answer factual questions about it [88, 89]. In other words, LLMs struggle to memorize facts and therefore to answer questions about entities mentioned less frequently in input data. All of these attest to the time-tested but underrated knowledge that the occurrence of query terms in a text is a strong indicator of a document's relevance to the query [90].

The CCR task comes with a pre-selected set of entities, but opportunities to expand it to discovering emerging entities can also be explored following the approach in [69]. Following [60], entity type information can be automatically used to build a type system that can be then used to force a neural network's output to respect the type system.

The CCR task has some aspects that do not seem amenable to neural network approaches. For example, in the distinction between citation-worthy and relevant documents lies the concept of novelty, that is, if a document contains new (novel) information that is not in the

entity profile, warranting the inclusion of the document in the entity profile's citation list. The neural network approaches do not seem suited for this aspect. Another disadvantage is that neural network approaches are much more difficult to implement, although the release of pre-trained models (such as BERT and GPT) has reduced this problem significantly. But pre-trained models also suffer from temporal degradation [91], i.e. either their knowledge becomes obsolete or they are not updated on new knowledge, although some studies have shown that this problem can be mitigated with retrieval augmentation [92]. Other than those, the neural network approach seems an interesting area to explore for the CCR task. They can also be combined with the string-matching approach such that the output of the string-matching system can be fed to a neural network model for ranking.

In our work on the impact of filtering, we showed that cleansing may reduce recall up to 21% by removing (partial) document content. Recent work on token-free neural models [93] operating directly on raw text also seems to support this finding. Specifically, as opposed to neural models operating on sequences of tokens corresponding to word or sub-word units, token-free neural models are more robust to noise and perform better on tasks that are sensitive to spelling and pronunciation. Both indicate that the preservation of information in the representation of entities can increase performance.

II

News Recommendation

7

Factors Influencing News Consumption and Recommendation

This chapter looks into factors that impact news consumption and recommendation. Specifically, it looks into geographical proximity and how it relates to news consumers of a certain location and certain categories of news items. Following that, the chapter investigates recommender systems in a real-world setting. The chapter starts with a descriptive study on the influence of geography in news consumption and then proceeds to investigate online comparison of news recommender systems, some of which incorporate the findings from the descriptive study into their recommendation algorithm.

7.1. The Role of Geographic Information in News Consumption

We investigate the role of geographical proximity in news consumption. Using a month-long log of user interactions with news items of ten information portals, we study the relationship between users' geographical locations and the geographical foci of information portals and local news categories. We find that the location of news consumers correlates with the geographical information of the information portals at two levels: the portal and the local news category. At the portal level, traditional mainstream news portals have a more geographically focused readership than special interest portals, such as sports and technology. At a finer level, the mainstream news portals have local news sections that have even more geographically focused readership.

7.1.1. Introduction

Online news reading is increasingly becoming the norm, with traditional newspapers moving to online service provision and new news portals and aggregators emerging. One problem

with online news provision and consumption is the overwhelming number of news items available to consumers. It is in the interest of news providers and news consumers to mitigate this overload. This has resulted in the emergence of news recommender systems, systems that attempt to solve the overload by proactively recommending the news items that are deemed interesting to the news reader. The success of a recommender system depends on the understanding of the factors that affect news consumption. This includes understanding both the content of the news items and the behaviors and preferences of news consumers.

These factors can be categorized into content and non-content. Content factors are modeled by key-words and named entities [94], and topics [95]. Non-content factors include, among others, the user's current context, social media annotations and other subtle features. Social media annotations affect both user's news consumption and satisfaction [96]. Branded companies and friend annotations and recommendations increase both consumption and satisfaction [96]. The subtle features such as readability, writing style, the type of story, visual complexity, and use of photography also influence a user's decision to read a news item [97]. It has been shown that non-content factors are as competitive as content-based factors in influencing the user's decision to read news items [97]. However, to the best of our knowledge, we have not seen a study on the effect of geographical proximity on news consumption.

This study investigates the role of geographical information in news consumption. It is a descriptive analysis work with the goal of assessing the role of geographical information in news consumption and seeing its potential for news recommendation. Recently, item recency has been shown to be an important factor in news recommendation [98]. Together with geographical information, these spatio-temporal features may be attractive because they are easy to implement and computationally efficient.

Using a dataset of user interaction with news items during a one-month period, we analyze and quantify the role of users' and items' geographical information in news consumption. Analysis is done at two levels: the information portal level and the local news category level. The contributions of this chapter are as follows: 1) Analysis and comparison of information portals based on geographical distribution of their news readers. 2) Investigation of the local and non-local news categories of mainstream portals with respect to the geographical distribution of their readership and 3) Describing and quantifying the role of the relationship between geographical information of mainstream information portals (and their local and non-local categories) and user's geographical location on news consumption.

7.1.2. Data

We use 53 million user-item interactions with items of 10 information portals collected by Plista[1], over a period of one month. Plista provides the Open Recommendation Platform (ORP), a framework that brings together information portals and news recommendation providers (referred to as participants). When a user starts reading a news item, a recommendation request is sent to one of the participants while the other participants receive the impression information. Every participant has access to all user-news item interaction information. The logs have been annotated by Plista with URLs of news items and state-level postcodes of news consumers. From the URLs, we can detect the local news items (as opposed to the non-local news items). The state level postcodes represent the user's geo-

[1]https://web.archive.org/web/20160514064752/http://orp.plista.com/documentation, accessed in July 2015

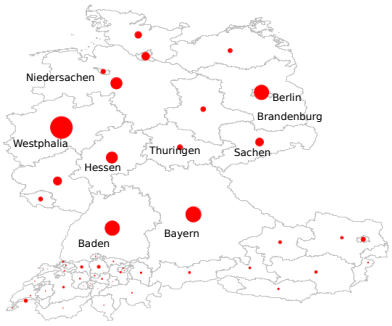


Figure 7.1: Bubble map of all users across states. Most users come from German states and Westphalia produces the largest number of users.

graphical location and the local and non-local news categories represent geography of the news items.

The Information Portals

Table 7.1 presents the information portals, their URLs and types. Figure 7.1 presents the distribution of the total number of users in our analysis by states. Most users come from Germany, and the state of Westphalia produces the highest number of news readers, consistent with the fact that it is the state with the single largest population. Figure 7.2 presents the distribution of users by portal. The automotive forum (Motor-talk), the two mainstream news portals (KStA and Tagesspiegel) and the sport news portal (sport1) have larger readerships. Two of the ten portals (Tagesspiegel and KStA) are traditional mainstream news portals providing opinion, politics and current events, and they can be national or regional. The other portals are special interest portals focused on information technology (4), sports news (1), automotive (1), business (1) and home and gardening (1).

Users and Items

Using cookie identifiers for user identification has a shortcoming in that, if a user does not maintain a persistent account, s/he will be counted more than once. Items are identified by unique numerical identifiers. Both items and users have many attributes. For our analysis, we focus on the state-level postcodes of users and on the URLs of news items.

User Location Information: States Our analysis is focused on the 52 states of Germany, Switzerland and Austria for two reasons. The first reason is Plista provided us with the mapping to the real postcodes of only the three countries' proxy postcodes that are originally used to represent the different states. The real postcodes help us anchor and contextualize our findings to actual geographical locations. The second reason is that the states of the three countries are geographically close to each other, German-speaking (all the information portals are in German language) and thus of primary interest for our study.

Item Location Information: (Non-)local News The two mainstream news portals organize their content in different sections of which city columns have our special interest, as

Table 7.1: The information portals. The short names are the names by which we refer to the portals in plots.

Short name	Type	URL
Cfo	Business	cfoworld.de
Cio	IT News	cio.de
Woche	IT News	computerwoche.de
Gulli	IT& Games	gulli.com
KStA	News	ksta.de
M-talk	Automotive	motor-talk.de
Channel	IT	tecchannel.de
Sport1	Sports	sport1.de
Tage	News	tagesspiegel.de
WH	Garden	wohnen-und-garten.de

news items deemed geographically relevant to the particular cities are placed under them. Tagesspiegel has the Berlin column (www.tagesspiegel.de/berlin) and KStA has the Cologne column (www.ksta.de/koeln) as their respective local news. We take advantage of the manual placement of news items (by the news editors) into the respective local news sections as a manual geotagging process. We consider all the news items that fall under a city column as local news and all the rest as non-local. We identify two subsets for Tagespiegel: Berlin (**Berlin**) which is the local news category, Tagesspiegel-minus-Berlin (**Tages-Berlin**) which is all the news that are not under the local category. We do the same for KStA: Cologne(**Cologne**) and KStA-minus-Cologne (**KStA-Cologne**). For comparison, we also include Tagesspiegel's sport section (**Tages' Sport**).

7.1.3. Analysis and Discussion

We analyze the information portals, with a view to finding similarities and patterns in the geographical distribution of their readership. Then we analyze the mainstream news portals and their local news categories also for similarities and differences in geographical distribution of their readerships. In both cases, we first aggregate the readers of an information portal or local categories by the 52 selected state-level postcodes. From the aggregated counts, we compute geographical likelihood distributions (across the states) of the readerships of the information portal or the local categories. Then we employ the Jensen-Shannon Distance (JSD) metric to quantify the difference between the geographical likelihood distributions (Equation 7.1). The uppercase letters X and Y represent vectors of likelihood distributions and KL stands for Kullback–Leibler divergence (Equation 7.2). Note that JSD is the square root of the Jensen–Shannon divergence, and it is a distance metric: the smaller the distance score between two likelihood distributions, the more similar they are. Finally, we examine and analyze how well we can correctly predict the likelihood of a user's state given the portal or the local news category the user visits.

$$JSD(X,Y) = \sqrt{\frac{1}{2}KL(X, \frac{(X+Y)}{2}) + \frac{1}{2}KL(Y, \frac{(X+Y)}{2})} \tag{7.1}$$

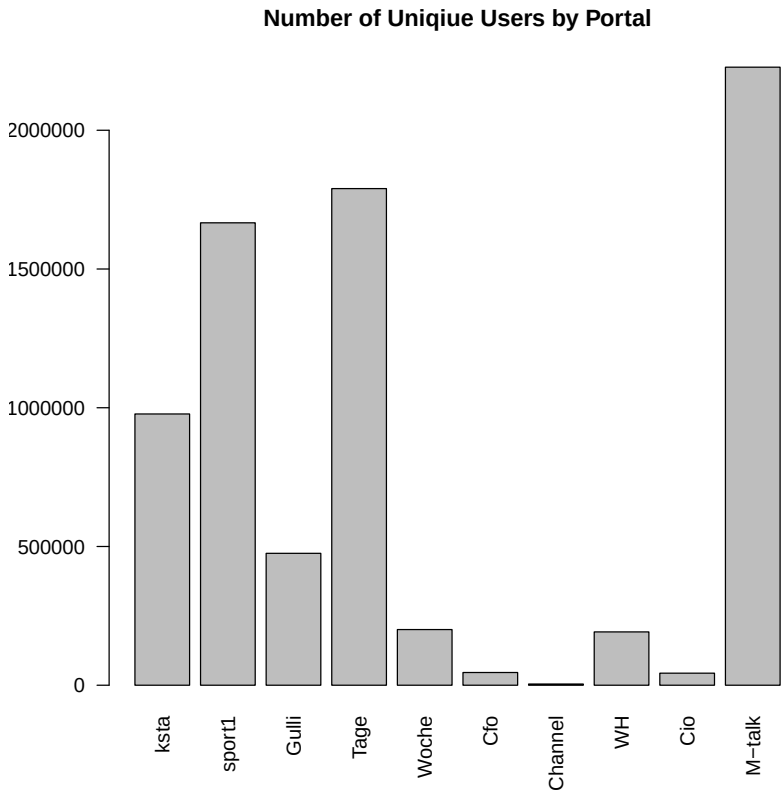


Figure 7.2: User frequency distribution by information portal.

$$KL(X,Y) = \sum_i x_i \ln \frac{x_i}{y_i} \tag{7.2}$$

Mainstream vs. Special Interest Portals

We characterize information portals by geographical distribution of their readerships modeled using conditional likelihood $P(user\ state\,|\,portal)$. Using JSD between the conditional likelihood distributions, we can determine how geographically similar their readerships are. The results are presented in Table 7.2. Firstly, the highest JSD observed between any two portals is 0.368, that is between KStA and Tagesspiegel. Secondly, we observe that the first and the second highest JSD scores of every special interest portal are from KStA and from Tagesspiegel respectively (see the colored columns and rows in Table 7.2).

The first observation tells us that the mainstream news portals differ the most in geographical readership. The second observation indicates that the two mainstream news portals have geographical user distributions that are very different from those of the special interest portals. Together, these observations indicate that, even in an online world, mainstream news portals are perceived as representing a certain geographical region and their readerships are mainly from those regions, while special interest portals are not bound to a geographical region of the type mainstream news portals are. The JSD scores between each of the special interest portals are small compared to the JSD scores between special interest portals and mainstream news portals. The distance scores between every special interest portal and mainstream news portals vary from 0.187 to 0.330, whereas the distance scores between each of the special interest portals vary from 0.033 to 0.140.

Figure 7.3 presents bubble maps of the user frequency counts of each state for the two mainstream news portals, and for two examples of special interest portals, for comparison. The bubble maps for the mainstream news portals have geographical foci. KStA's readership is mainly from its home-state (Westphalia) and Tagesspiegel's readership is more distributed than KStA's. The bubble maps for the two special interest portals (Sport1 and Gulli), however, are more evenly distributed. These observations are indications that there is an association of mainstream news portals with some geographical focus while the appeal of interest portals seems not to be limited to the same geographical constraint. The mainstream news portals are interesting for the following additional reasons too. First, they are two of the three portals that receive significant clicks on recommended articles [98]. Second, they are the portals that offer the opportunity for extracting geographical local and non-local news categories, which we discuss later below.

Local vs. Non-local News Categories

For each local news category, users are aggregated by state-level postcodes. Then we compute $P(user\ state|locale)$ which is a geographical likelihood distribution (across the states of the three countries) of the local news readership. Using the geographical likelihood distributions, we compute JSD scores between the four news categories. The results are presented in Table 7.3. The highest distance observed (0.561) is between Berlin category (Tagesspiegel's local news) (**Berlin**) and Cologne category (KStA-koeln's local news (**Cologne**)), an indication that the geographical distributions of their readerships are the most different. The next highest distance observed (0.485) is between KStA and **Berlin**. Tages-Berlin and KStA-Cologne reprsent the non-local categories of each.

Table 7.2: Adjacency matrix of information portals based on the Jensen-Shannon distance between the geographical distribution of their readerships. The highlights show the distances between special interest portals and mainstream news portals.

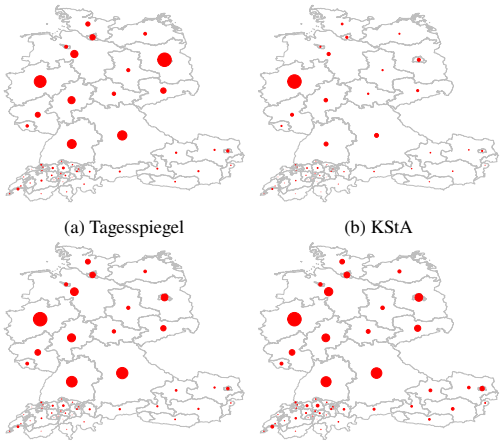
	WH	M-talk	Tage	Woche	Cio	Cfo	Channel	KStA	Sport1
Gulli	0.067	0.057	0.187	0.066	0.101	0.129	0.043	0.322	0.102
Sport1	0.099	0.080	0.192	0.091	0.105	0.131	0.119	0.305	
KStA	0.330	0.314	0.368	0.323	0.321	0.332	0.331		
Channel	0.067	0.062	0.209	0.055	0.087	0.111			
Cfo	0.140	0.127	0.229	0.082	0.053				
Cio	0.110	0.093	0.215	0.044					
Woche	0.076	0.060	0.198						
Tage	0.221	0.210							
M-talk	0.033								

It is interesting to examine the differences between the different categories of news items published in the same portal. This means the distances between Tages, **Berlin**, **Tages-Berlin**, **Tages' Sport** on one hand, and KStA, **Cologne** and **KStA-Cologne** on the other. The distance between **Berlin** and Tages is 0.200, whereas the distance between **Tages Sport** and Tages is 0.038. Clearly, this shows how geographically different the readership of the Berlin category of local news is from the readership of the full portal or its sports' section. The distance between Berlin and Tages-Berlin is 0.230, indicating further that the Berlin category of local news has a geographically more distinct readership. It is also important to compare the difference with the distance between Cologne and KStA-Cologne which is 0.133. The almost double distance between the local and non-local sections of Tagesspiegel is an indication that the Berlin category of local news and Tagesspiegel-Berlin have a large difference in geographical readership distributions. We explain this by the fact that Tagesspiegel has a wider readership that covers a larger geographical area. We interpret the smaller distance between KStA and KStA-Cologne as evidence that KStA reaches a narrower geographical readership anyway, that is that KStA has a more regional character.

Our explanation of viewing Tagesspiegel as a national newspaper as opposed to KStA as a regional one is supported by the bubble maps of Figures 7.4 and 7.3. In Figure 7.4, we clearly see that the readership of the Berlin category of local news and Cologne category of local news is more geographically localized than that of Tagesspiegel-minus-Berlin and KStA-minus-Cologne, and that the readership of the Cologne category of local news is more localized than that of the Berlin category of local news.

Likelihood of a User's State Given a Portal or a News Category

Another way to look at the relationship between a user's geographical location and a geographical information of portals and categories is to compute the likelihood of correctly predicting the user's state given the information portal (or category) and a cutoff value of user's visit frequency. Specifically, we compute the likelihood $P(\mathit{user\ state} \mid \mathit{portal}, \mathit{cutoff})$ and $P(\mathit{user\ state} \mid \mathit{locale}, \mathit{cutoff})$. For the special interest portals, since their readerships are geographically distributed, the likelihood of predicting a user's state correctly is very low (less than 0.2). Therefore the likelihood of predicting a user's state from their visits of portals and news categories is interesting only for the mainstream news portals and their



(a) Tagesspiegel (b) KStA

(c) Sport1 (d) Gulli

Figure 7.3: Bubble maps of the state-level distribution of users for the mainstream news portals (Tagesspiegel and KStA) and two specialized portals (Sport1 and Gulli). KStA has a very localized readership. Tagesspiegel and the special interest portals show a more distributed readership.

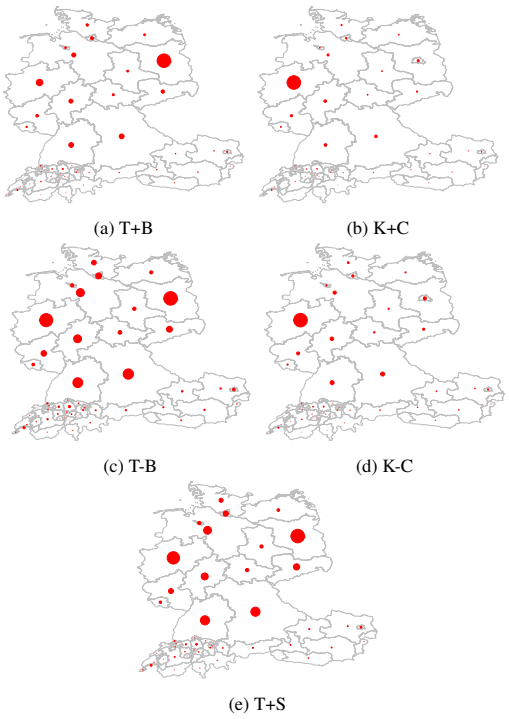
local and non-local news categories.

For the mainstream portals, the likelihood that a visiting user is from the state of their geographical focus is very high (as compared to the likelihood that the user is from any of the other states). Therefore, we focus on the likelihood of the respective geographical focus for each of the mainstream news portals and their local and non-local categories. The results are presented in the plots of Figure 7.5. We observe that the likelihood that a user reading a Cologne category of local news is from the state of Westphalia is as high as 0.8, as compared to a user reading KStA which gives the likelihood of 0.40. In the case of the Berlin category of local news, the likelihood that a user is from the state of Berlin is 0.48 and in the case of Tagesspiegel, the likelihood that a user is from the state of Berlin is 0.22.

As we increase the cutoff of the frequency of visits of the user, we observe that the likelihood of predicting a user's state increases. The gap between the plots of the Berlin category of local news, and the Tagesspiegel-minus-Berlin category is a measure of the strength of

Table 7.3: Adjacency matrix for the local and non-local news categories based on Jensen-Shannon distances between the geographical distributions of their readership. Note the largest distances between K+C and T+B, and between T+B and KStA. Note also the large difference between the distance scores of T-B and T+B (0.230), and between K-C and K+C (0.133).

	Tages	KStA	Berlin	Cologne	KStA-Cologne	Tages-Berlin
Tages' Sport	0.038	0.360	0.207	0.465	0.358	0.046
Berlin	0.031	0.354	0.230	0.465	0.351	
KStAa-Cologne	0.366	0.003	0.483	0.133		
Cologne	0.474	0.130	0.561			
Berlin	0.200	0.485				
KStA	0.368					

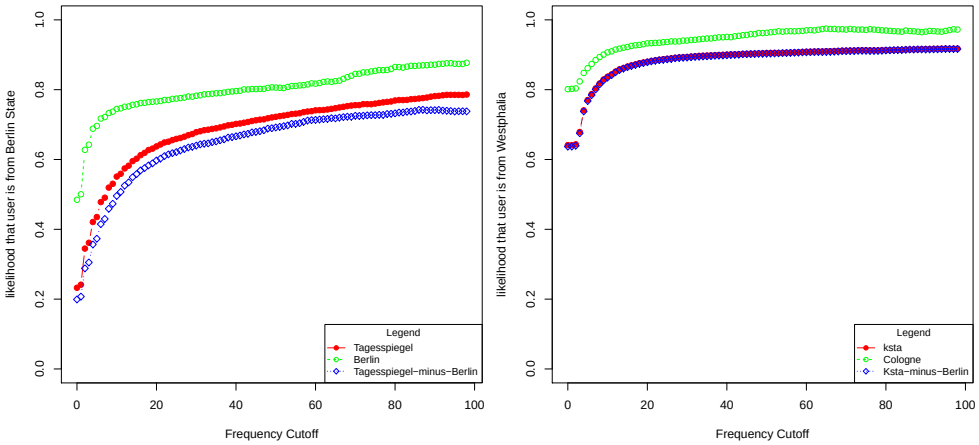


(a) T+B (b) K+C

(c) T-B (d) K-C

(e) T+S

Figure 7.4: Bubble maps for the state-level distribution of the readership of the news categories of the two mainstream news portals. Note that Tagesspiegel's Berlin local news section has very different geographical readerships from Tagesspiegel-minus-Berlin and Tagesspiegel' sport section. Note also that Cologne and KStA-minus-Cologne have almost the same geographical readership.



(a) The likelihoods of a user being from the state of Berlin for local and non-local news categories of Tagesspiegel.

(b) The likelihoods of a user being from the state of Westphalia for the local and non-local news categories of KStA.

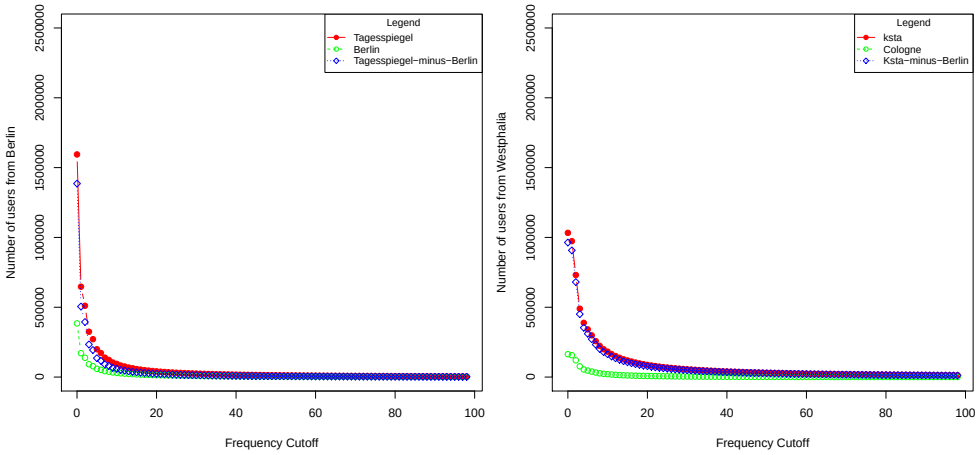
Figure 7.5: Each figure presents the $P(Berlin\ state \mid locale, cutoff)$ and $P(Westphalia \mid locale, cutoff)$ for the news categories of Tagesspiegel (7.5a) and of KStA (7.5b) respectively. We see a wider gap between the plots of Berlin and Tagesspiegel than between Cologne and KStA, an indication of difference in geographical readerships of Berlin and Tagesspiegel from Cologne and KStA. We also see that the plots of KStA-minus-Cologne and KStA overlap because the number of user-item interactions for Cologne is very small compared to KStA-minus-Cologne.

the geographical information in the local news consumption. On the categories of KStA, however, the gap between the Cologne category and KStA-minus-Cologne is small, indicating a more or less the same readership for the local news and the portal itself. It is also worth noting that the readership of the Cologne category is small compared to the readership of the KStA-minus-Cologne, and has no impact on the combined plot (KStA). Our explanation for the difference in likelihoods of predicting the respective states for Cologne category of local news and Berlin category of local news is that, by virtue of the state of Berlin being the capital, it attracts users from all over the country, more than Cologne does.

Discussion

We observe that geographical information plays an important role in user's consumption of news items of the mainstream news portals, and that it manifests itself at two levels: the portal level and the local news categories level, as can be observed from tables 7.2 and 7.3. The bubble maps of Figures 7.3 and 7.4 visually confirm these observations. Geographical information at the portal level manifests itself in the sense that users associate a strong or loose geographical location to the portal itself. This finding may be useful in news aggregators (such as Google News and Yahoo! News) to identify news publishers that are geographically relevant to certain users.

The second level where geographical information manifests itself is at the local and non-local categories. Such fine-grained geographical information is useful, for example, for tailoring recommendations for the local and non-local news visitors. Together with associated geographical focus of the portal, the local and non-local categories may be used for improv-



(a) The number of users for the news categories of Tagesspiegel.

(b) The number of users for the news categories of KStA.

Figure 7.6: Each figure presents the number of users remaining versus cutoff values for the news categories of Tagesspiegel (7.6a) and of KStA (7.6b).

ing news recommendation. We imagine that such geographical information can be useful in big countries where there are competing national and regional news portals.

7.1.4. Conclusion

We have investigated a dataset of one month of user interactions with news items of different information portals. We measured the distance, based on geographical distribution of readerships, between different news portals and found out that mainstream news portals and special interest portals show differences in the role geographical information plays in influencing users. While the special interest portals seem to be less geographically localized, the mainstream news portals, on the other hand, exhibit geographical foci. The mainstream news portals were further analyzed by focusing on their local news categories which also showed a more localized geographical readerships. We showed the likelihood that a user is from the home-state (the geographical focus) of the mainstream news portal can be predicted reasonably well, especially when higher cutoff values of the user's visit frequency are considered. The relationship between the geographical location of news users, and the geographical foci of mainstream news portals and their local news categories can be exploited for improving news recommendation, which is our future work.

7.2. Real World News Recommendation

In the previous section, we have investigated the role of geographical information in news consumption. In this section, we deploy algorithms in a real world setting to understand the performances of our algorithms, to apply the geographical information in a real world recommender system, and to understand non-algorithmic factors in real world recommender system evaluation.

We pursue three goals in this Section. The first goal is to compare our algorithms in a real-life recommendation setting. We accomplish this by deploying four algorithms and comparing their performance behaviors. The second goal is to see whether the incorporation of geographical information into news recommendation results in an improved performance. In the descriptive study we conducted on a Plista dataset in our pre-2015 participation [99], we reported two findings. One is that there is a substantial difference in the geographical distribution of the readerships of traditional news portals, and the second is that within the same news portal, the geographical distribution of the readerships of the local news category and the rest of the categories shows substantial difference. The third goal is to introduce a way to measure real world non-algorithmic influences on real world algorithms by deploying two instances of the same algorithm.

This work was done in the context of CLEF NewsREEL's News recommendation evaluation [100] in 2015. CLEF NewsREEL is a campaign-like news recommendation evaluation initiative [101], that provides opportunities to investigate recommender system performance from several angles. CLEF NewsREEL provides the Open Recommendation Protocol[2], a place where recommendation providers can plug their algorithms to provide recommendation for news portals in need of recommendation. CLEF NewsREEL had two tasks [20]: Benchmarking News Recommendations in a Living Lab (Task 1) and Benchmarking News Recommendations in a Simulated Environment (Task 2). Benchmarking News Recommendations in a Living Lab (Task 1) enables the evaluation of recommender systems in a production setting. This has the advantage of testing algorithms in a real-world setting where recommendations are bench-marked by actual users. Benchmarking News Recommendations in a Living Lab is challenging because of the technical aspects of time constraints, responsiveness and scalability. Additionally, continuous change of items and user preferences add to the challenge. Benchmarking News Recommendations in a Simulated Environment (Task 2) enables the evaluation of systems in a simulated (offline) setting using dataset collected from the online interactions. This allows participants to evaluate their recommender systems without the constraints of time, scalability and responsiveness. Task 2 is reproducible, and suited for fine-tuning parameters and optimization. In 2015, we participated in Task 1.

7.2.1. Approach

We devised several simple but effective algorithms. Among our algorithms, we included two instances of the same algorithm, with the objective to measure the differences in performance that would have to be attributed to randomness - differences between distinct instances of the exact same algorithm, deployed in the same online recommendation scenario, during the exact same period of operation. A direct comparison of the results that should be identical provides us with the opportunity to consider one instance as the baseline, and obtain a quantitative measure of the performance difference that could only originate from non-algorithmic factors.

7.2.2. Experiments

We experimented with five algorithms, all of them modifications of a straightforward approach to recommendation based on *recency*. The recency algorithm takes into account

[2]https://orp.plista.com/, accessed in July 2015

recency and popularity of an item, and it has been shown to be a strong baseline in previous online evaluations. The algorithmic variations that we experimented with are listed below.

Recency: This algorithm keeps the 100 most recently viewed items for each publisher in consideration for being recommended to the user. The most recently read items are returned in response to a recommendation request. We run two instances of this algorithm to get a sense of the randomness involved in the selection of algorithms by the Plista framework [102] and/or clicks on recommendations by users.

RecencyRandom: Instead of recommending the five or six most recently viewed items, this approach returns a random selection from the top 100 most recently viewed items.

GeoRec: The geographical recommender takes the geographical region (states to be specific) of users and the local category of news items into account when generating recommendations. We generate two sets of recommendations, one by the recency recommender and one by a purely geographical recommender. For the purely geographical recommender, we take the 100 most recently viewed items and sort them according to their geographical conditional likelihood scores generated by Equation 7.3, which is the likelihood that a user from a given state reads news from a given category.

$$r_{u_a,i_k} = P(c_{i_k}|g_{u_a}) \tag{7.3}$$

where c_{i_k} is a category label corresponding to the local category of item i_k (local or non-local) and g_{u_a} is the state-level geographical information of the user u_a, that is, the state the user belongs to. In effect, for a given user, this recommender recommends items either from the local category or from the non-local category. We combine geographical recommendations with recency recommendations as follows. First, we intersect twice the number of recommendations requested from the geographical recommender with the requested number of recommendations from the recency recommender. If the resulting set is smaller than the number of recommendations requested, we append $half - 1$ of the needed (but not filled) items from the geographical recommender and another $half + 1$ from the recency recommender.

GeoRecHistory: This modification of the GeoRec recommender excludes items that the user has already visited from recommendation.

7.2.3. Results and analysis

We ran the recommendation systems in Section 7.2.2 for a period of 86 days, between April 12th and July 6th, 2015 with two exceptions; the RecencyRandom algorithm was started 12 days later, on April 24, 2015, and the geoRecHistory ran only for the first 53 days. The two geographical recommender systems (GeoRec and GeoRecHistory) are supposed to study the exploitation of geographical proximity for the improvement of recommender system performance, in addition to being used for the examination of recommender systems from multiple dimensions.

The Click-through Rate (CTR) scores are presented in Table 7.4. We see that the performance differences are small. If we would rank the algorithms based on their performance, however, the GeoRec recommender leads, followed by Recency, then by Recency2, then GeoRecHistory and finally by RecencyRandom. Figure 7.7 shows cumulative CTR as a function of the number of days, for the same period. The cumulative captures the overall CTR performance for all the days, as opposed to isolated daily performances.

Table 7.4: Number of requests, number of clicks and CTR scores of five systems in 2015. Except GeoRecHistory (which ran for 53 days), the other four ran for 86 days.

Algorithm	Number of requests	Number of Clicks	CTR (%)
Recency	90663	870	0.96
Recency2	88063	810	0.92
RecencyRandom	73969	596	0.80
GeoRec	88543	847	0.96
GeoRecHistory	47,001	395	0.84

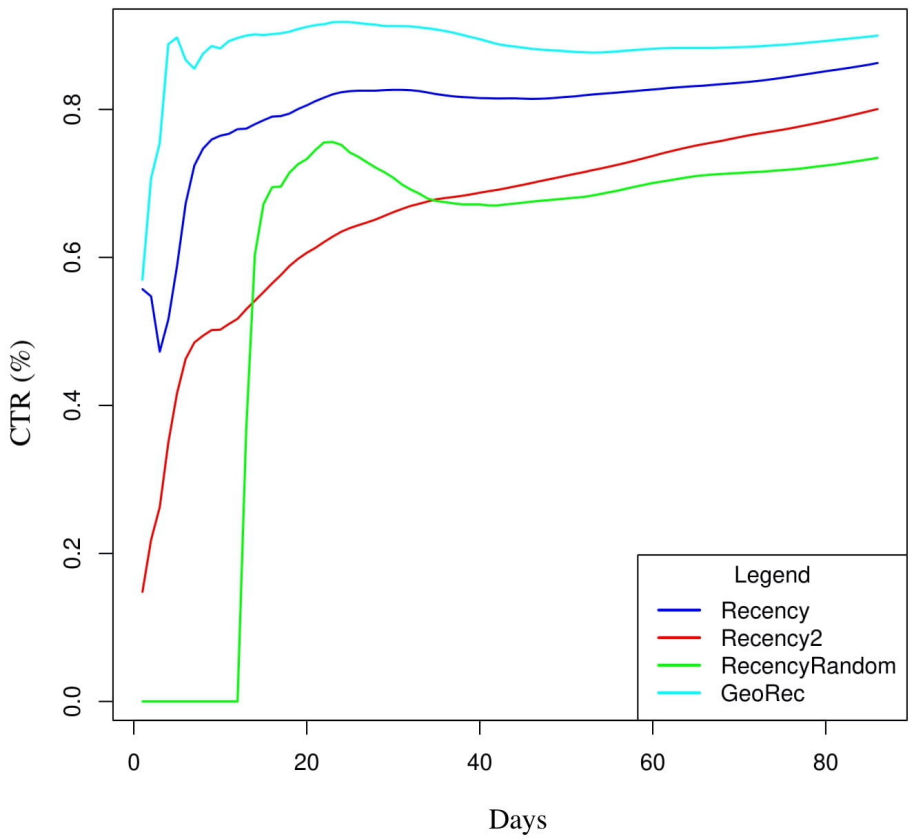


Figure 7.7: The cumulative CTR performances of the five algorithms as they progress on a daily basis. GeoRecHistory is excluded as it didn't run for the entire period. RecencyRandom started 12 days later.

Table 7.5: Checking statistical significance daily for a period of 86 days using RecencyRandom as a baseline. GeoRecHistory's results are based on 53 days.

Algorithm	Days of Significance	Percentage (%)
recency	20	27.4
geoRec	41	56.2
geoRecHistory	24	60.0

Table 7.6: Checking statistical significance daily for a period of 86 days using Recency2 as a baseline. GeoRecHistory's results are based on 53 days.

Algorithm	Days of Significance	Percentage (%)
recency	2	2.7
geoRec	25	34.3
geoRecHistory	1	2.5

Comparison of the Recommender Systems

From the cumulative plots (Figure 7.7), we see that the performance measurements vary considerably. We see that the results for Recency and Recency2 differ considerably during a large part of the evaluation period, although, eventually, converging to a stable situation. If one were to continuously monitor the measured performance of the two instances, one might easily conclude (wrongly) that Recency is a better approach to recommendation than Recency2. Imagine for example an experimenter peeking at the experiments every day, to make a decision as to which is the best among the competing instances. How many times would the experimenter declare statistically significant differences between the different instances? We examined this by using two baselines: the random recommender (RecencyRandom) and Recency2. The results when using the RecencyRandom recommender as a baseline are given in Table 7.5. Similarly, the results for the baseline of Recency2 are given in Table 7.6. We see that, when RecencyRandom is used as the baseline, Recency, GeoRec and GeoRecHistory achieve significantly different performance for a majority of the days tested. With Recency2 as the baseline, we see that these percentages are lower; the difference with Recency is considered significant, at significance level 0.05, according to the test on two days.

The two instances of the same algorithm show large enough differences in performance that there is a chance of concluding one is better than itself. This observation raises questions regarding interpreting the results of online evaluation using A/B testing; it is not so easy to conclude that one algorithm is better than another based on just an observed difference in performance, even if a statistical test supports that decision. Given the dynamic nature of user-item interactions and the resulting differences in the particular settings that the algorithms operate in, we should be careful when interpreting a small but seemingly significant performance difference. Recommendation evaluations that involve user-item interactions must account for some level of randomness, and perhaps a more strict level of statistical significance should be considered than the commonly used 5%.

Impact of Geographic Information on Recommender System Improvement
The incorporation of geographical information into a recommender systems was realized by modifying the recommender system such that users belonging to the geographical focus of a news portal were also served recommendations about that geographical focus. In the algorithm of GeoRecHistory, already read items were excluded from being recommended again.

The results in Table 7.4, the cumulative CTR performances in Figure 7.7 and the significance tests show the incorporation of geographical information resulted in some performance improvement[3]. Compared to Recency, this is not a big improvement, but, as we will see in the next chapter, the geographical recommender did not modify the Recency recommendations list that much.

7.2.4. Conclusion

We set out to study the performance differences of live recommender system algorithms. We also specifically wanted to investigate the role of geographical information in real world news recommendation, and the effect of non-algorithmic factors in news recommendation evaluation. The algorithms are interrelated, as they are all based on recency, and yet their performances vary a lot. The geographical recommendation showed some performance improvement, the effect of randomness seems to indicate that care must be taken to take into account some degree of randomness in recommender systems evaluation that involve users in a live setting, perhaps a higher p-value for statistical significance tests.

[3]In our 2015 report [99], we reported there was no improvement, but we corrected this in our 2016 report saying that there was a mistake in calculation

8

Multidimensional Examination of News Recommendation Evaluation

8.1. Introduction

The purpose of most recommender system evaluations is to select algorithms for use in a production setting. Recommender Systems are evaluated in different settings and manners. Two common settings for evaluations are offline and online. Offline evaluations test the effectiveness of recommender system algorithms in a controlled environment by using a static dataset and a metric. Offline evaluations are easier and reproducible and they are usually assumed to reflect online performances. But do offline evaluations predict online performance behaviors and trends? Do the absolute performances of algorithms offline hold online too? Do the relative rankings of algorithms according to offline evaluation hold online too? How do offline evaluations compare to and contrast with online evaluations?

Real time news recommendations must meet the challenges of contextual relevance, dynamic item sets, dynamic user needs, and must also satisfy non-functional requirements such as response time and scalability [7]. A common approach to online evaluation evaluates recommender systems by a method called the A/B testing, where a part of users are served by recommender system A and the other part by recommender system B. The recommender system that achieves a higher score according to a chosen metric is regarded as a better one, given other factors such as latency and complexity are comparable. What is the validity of this type of evaluation and what are the challenges in it?

Another dimension in the evaluation of recommender systems online is time. Does the time in which the algorithms are tested have an impact in their performances or does performances hold across time? Also when algorithms are evaluated in a platform, how fair are the comparisons? This chapter deals with the challenges of the evaluation of News Recommendation across several dimensions.

The work in this chapter, like in the one in Section 7.2 of Chpater 7, is based on experiments conducted during our participation in the CLEF NewsREEL initiative. It combines our findings from 2015 and 2016. In 2015, we participated in Task 1. In 2016, we participated in both Task 1 and Task 2, as the focus of the initiative in 2016 was on comparing recommender system performance in online and offline settings [7].

We pursue several goals in this chapter. The first goal is to examine the causes of (random) performance differences between online news recommender systems. In the evaluation of recommender systems, the quality of recommendations made by a newly proposed algorithm is compared to the state-of-the-art, using a given metric and dataset. Validity of the evaluation depends on the assumption that the evaluation does not exhibit artifacts resulting from the process of collecting the dataset. The main difference between online and offline evaluation is that in the online setting, the user's response to a recommendation is only observed once. We attempt to quantify the expected degree of variation in performance that cannot be attributed to differences between systems. We classify and discuss the non-algorithmic causes of observed performance differences.

The second goal is to examine news recommender systems evaluations along several dimensions, namely offline, online, time, and non-algorithmic factors suchs as randomness and system (platform) idiosyncrasies by using an A/A test, which is running two instances of the same algorithm at the same time.

The third and final goal is to appraise, from a participant's (our) perspective, the CLEF NewsREEL News Recommendation Evaluation initiative, the platform under which the research in this chapter and the previous chapter are conducted. This section comes from our contribution to a collaborative work that attempts to appraise the CLEF NewsREEL initiative from multiple dimensions in order to draw meaningful lessons for better news recommendation evaluation.

The chapter starts with a recap of the performances of the two instances of an algorithm that we discussed in the previous chapter. The chapter then proceeds to investigate the causes of (random) performance differences in real-world recommender systems, followed by examination of recommender systems from multiple dimensions (time-wise, offline, online and A/A), and ends with an appraisal of the CLEF NewsREEL: News Recommendation Evaluation Lab platform.

8.2. Performance Differences Between Online Recommender System Algorithms

The literature on recommender systems shows that offline and online recommender system evaluations may not concur with each other [103–105]. Recommender systems may behave differently in offline and online evaluations, both in terms of absolute and relative performance. This has a serious implication for recommender system research, because the point of offline evaluation is the assumption that at least the relative performance of recommender systems is indicative of their relative online performance and thus an important step for selecting algorithms that can be deployed in a live recommendation setting.

Prior literature has pointed out a variety of explanations for the performance discrepancy between online and offline evaluations [105, 106]. First, offline evaluations can only measure accuracy in a static manner, leaving out the differences resulting from actual user

behavior. Naturally, offline datasets provide only an incomplete and imprecise model of the real world. The abstraction from user behavior and context by taking a snapshot of recommendations and user responses may deviate too much from reality to allow for a valid comparison between different recommender systems.

The online evaluation of recommender systems overcomes some of these limitations, because we can observe the actual users' responses to recommendations originating from a specific system. A drawback of this setup, however, is the additional "randomness" as a result of the dynamic nature of user-item interaction, the difference in the items served for the algorithms being compared, and the effect introduced by the system that orchestrates the distribution of items to recommender algorithms and the presence of non-functional factors affecting algorithms. These randomness needs to be accounted for for fairer and correct evaluation and comparison of news recommender system. The research presented here attempts to improve our understanding of how to accommodate for random differences, and still make the right inferences from the evaluation data obtained in CLEF NewsREEL. To identify factors that may explain observed performance differences in online recommender system evaluation, we conduct an investigation of the results of the algorithms, *two of which are distinct instances of exactly same algorithm*, that we ran in Section 7.2. We use the experimental results obtained to quantify the effect of randomness in online evaluation on the measured performance.

As we presented in Section 7.2 of Chapter 7, we included two instances of the same algorithm, with the objective to measure the differences in performance that would have to be attributed to randomness – differences between distinct instances of the exact same algorithm, deployed in the same online recommendation scenario, during the exact same period of operation. A direct comparison of the results that should be identical provides us with the opportunity to consider one instance as the baseline, and obtain a quantitative measure of the performance difference that can only originate from non-algorithmic factors. By also logging the recommendation requests, responses, and clicks, we can recreate the recommendation scenario of one algorithm and compare its results to those that would have been given by the other algorithms. Mixing online and offline evaluation methods provides a more controlled way of measuring differences between different recommender systems, that we can use to estimate the part of the difference in performance that should be attributed to chance.

8.2.1. Causes of Performance Differences

We saw that the two identical instances of the Recency algorithm, Recency and Recency2, can at times differ considerably to the extent that one might even conclude that the Recency algorithm is a better approach to recommendation than Recency2. This observation raises questions into the validity of reported improvement in real world recommender system evaluations. The results on the exactly identical instances call for identification, quantification and explanation of the factors that cause "random" performances differences between identical instances of an algorithm. It is also interesting and useful if the quantification can be applied to quantify the "random" performance differences, as opposed to the differences from their unique strengths, between non-identical instances.

The random performance difference between identical instances is indicative of the extent of performance difference, in non-identical instances, that can arise due to non-algorithmic

factors such as receiving different user-item interactions from the evaluation framework. In a real world setting, the actual users and items that algorithms deal with differ from instance to instance. In the following subsections, we identify the non-algorithmic factors that may cause the differences in performance, explain them and propose a method to quantify them. We specifically distinguish 1) operational differences in the evaluation framework, 2) differences in user-item pairs for which recommendations have been provided, and 3) remaining differences that we consider due to randomness.

8.2.2. Operational Causes

By non-algorithmic operational causes, we refer to decisions in the evaluation framework that could affect the observed performance of the recommender systems evaluated. Recommendation systems under evaluation are served requests by a system that distributes the incoming requests in a supposedly fair manner. From the perspective of the CLEF NewsREEL participant, fairness of this process is a matter of faith, and difficult to assess. We know that some publishers are more likely to trigger clicks on recommendations than others, such that biases in the distribution of recommendation requests can easily result in performance differences between the algorithms under evaluation. The approach of assigning recommendation requests to participant systems may exhibit an (implicit) bias with respect to pairing some teams and/or systems with a subset of publishers, or assigning specific users (e.g., those logged-in) to some teams or algorithms, or serving a skewed subset of items from specific categories (e.g., political), or a combination of such factors.

8.2.3. User-Item Causes

Another source of differences in performance that are not algorithmic could arise due to differences in the sets of items and users that are assigned to the two instances. Every instance under evaluation receives a different subset of all recommendation requests, resulting in inherent differences in performance if, by chance, certain user-item interactions are incomparable (which would also render the measured results incomparable). In the evaluation of information retrieval systems, for example, it is well known that results obtained on different test collections cannot be compared directly; here, to some extent, we could consider the different performance measurements to result from different test collections, and direct comparison may suffer from the same problems as in the information retrieval evaluation case.

8.2.4. Random Causes

We refer to all remaining factors that might cause performance differences as random causes, including factors like the user's mood as well as causes that result from idiosyncrasies of the particular datasets (settings, in the online case). Imagine an offline setting with two algorithms (algorithm one and algorithm two) and two datasets (dataset one and dataset two). If on dataset one, algorithm one performs better than algorithm two, but on dataset two the situation is reversed, the difference between the performance measurements cannot be attributed to the difference in users and items or differences in the algorithms.

One of the advantages of running four instances of algorithms at the same time is that we have datasets that have one big advantage over disparate datasets used for research and that is that we have their online behavior and performance. These logs are, therefore, very

Table 8.1: Shared recommendations. The score in each cell is the percentage of the lists that the two recommendations had in common, and the second number, between brackets, is a percentage of the sets of recommendations that the algorithms had in common. GeoRec-Recency2 and GeoRec-Recency show the highest similarities.

Algorithms	Recency	GeoRec	RecencyRandom
Recency	100	**85.82(97.96)**	0.0(74.11)
Recency2	100	85.79(97.97)	0.0(73.84)
GeoRec	50.99(91.64)	100	0.0(76.18)
RecencyRandom	0.01(73.28)	0.01(73.40)	100

important to the performance difference that arises as a result of the random causes in an online setting, as discussed below.

8.2.5. Overlap in Performance

How can we find out that the random causes (idiosyncrasies of the particular setting) are having an impact on the performance differences of algorithms? To measure the effect of artifacts in evaluation data on performance estimates in an offline setting, we could evaluate two different algorithms on two datasets, and measure the performance differences between the algorithms on each individual dataset. The absolute difference between these two differences can be considered an estimate of the "dataset artifact" on performance. For, if there is no difference, then the measurements are accurate, and both datasets lead to the same conclusions. However, if a difference is observed, then we would seek the cause for these variations in the differences between the evaluation data. In an online setting, it is not possible to follow this exact procedure, but it is possible to quantify a part of this dataset (setting) artifact using a similar method.

Imagine an ideal world where you can run two algorithms simultaneously in exactly the same environment. Users, items, and time are exactly the same. The only things that differ in this ideal world are the recommendations and responses by the algorithms. Table 8.1 shows how different (similar) the recommendation by other algorithms on the different settings would be. The scores are the percentages of shared recommendation over the total number of recommendations. The table gives two scores for each pair, the first being the exact similarity per recommendation response both in order and content (the number given in each table cell), and the other being the set similarity per recommendation response (order can vary) given between brackets. Each cell corresponds to the similarity measured when the algorithm listed in the column is applied to a dataset constructed from the log obtained when using the algorithm listed in the row. GeoRec-Recency and GeoRec-Recency2 show large similarities, which is not surprising since the GeoRec recommender is only a minor modification of the recency recommender. GeoRec is supposed to diversify Receny's results; but apparently it did not diversify the results much in practice.

The idealized system described above would enable us to determine, in the true sense, the algorithm that is the better one; at least, in the evaluation framework in which the algorithms in question are being tested. In practice, such a test would be an approximation, since it does not account to the many factors that can cause performance differences. Obviously such an

idealized system is hard to create, but we can create one aspect of that idealized system. That aspect is the overlap in performance that two algorithms would have if they were to be run in the idealized system. The overlap in performance is defined in Equation 8.1.

$$Setting_A Overlap_{AB} = \frac{Clicks_{AB}}{Recommendations_{AB}} \tag{8.1}$$

In Eq. 8.1, $Setting_A$ is the log generated by running algorithm A, and $Setting_A Overlap_{AB}$ is the overlap in performance of algorithms A and B in dataset $Setting_A$. $Clicks_{AB}$ and $Recommendation_{AB}$ are counted from intersection of recommended items and the intersections of recommended-and-clicked items respectively of algorithms A and B, when they would be run in an exact online setting that would generate $Setting_A$. The overlap in performance is the ratio of the intersection of recommended-and-clicked items and the intersection of recommended items that two online-deployed algorithms would share if they were to be run in the idealized system. We use this overlap in performance to quantify a part of the performance difference as a result of the random causes by comparing the overlap in performance of two algorithms in two datasets.

To explain how we would obtain the overlap in performance, consider the two algorithms which we used in the NewsREEL challenge. For each algorithm, we have logged the recommendation request, recommendation response, and clicks. If we rerun the other algorithm on the logs of the first algorithm, everything remains the same except the recommendation responses. By determining to what extent the recommendations are the same for the two algorithms, and the ratio of the clicks received by the online-deployed algorithm could also have been obtained by the competing algorithm running on the logs, we obtain the overlap in performance. To obtain the overlap in performance of two algorithms in the idealized system we described, one does not need to run both algorithms online. Running one algorithm online to obtain logs that form a dataset for evaluation, and subsequently running the other algorithm on these logs, is sufficient; for, it is only the overlap of the two algorithms that we are interested in, and not the overall performances of the algorithms.

Difference in Overlap

If we have two online-deployed algorithms and record both of their logs, we can determine a measure of overlap between the two algorithms on each of these logs. We call the difference between the two measures of overlap the **difference in overlap**, its definition given by Equation 8.2. Note that to compute this difference in overlap, we need to deploy both algorithms and collect their respective logs. If there are no differences in behavior of these algorithms on the same logs, this difference would be zero. The difference in overlap therefore gives us a measure that quantifies the overall difference in performance that should be attributed to non-algorithmic causes.

$$DiffinOverlap_{Setting_A Setting_B} = |Setting_A Overlap_{AB} - Setting_B Overlap_{AB}| \tag{8.2}$$

Since we have four algorithms that ran during the complete evaluation campaign (excluding GeoRecHistory), we can quantify differences in overlap between several pairs of algorithms, and, together, these differences in overlap will give us a clue of the extent to which performance differences between algorithms should be attributed to chance. In other

Table 8.2: Difference in overlap of our algorithms. Each entry is obtained by subtracting overlap in performance in one dataset of two algorithms from their overlap in performance in another dataset. GeoRec-Recency2 and GeoRec-RecencyRandom show the highest overlap difference.

Algorithms	Recency	Recency2	GeoRec	RecencyRandom
Recency	0	0	0.001	0.006
Recency2		0	**0.026**	0.004
GeoRec				**0.026**

words, even though the full difference in overlap cannot be measured, as we cannot create the idealized system where two different algorithms would receive the exact same recommendation requests for the exact same user and item combinations, by zooming in on the performance overlap we can still obtain an estimate of the level of non-algorithmic differences in the evaluation.

To calculate the difference in overlap, we make one assumption, and that is that we do not take into account the order of the recommended items. If two algorithms have recommended two lists of the same items, but in different order and a click happened on the online deployment, we consider a click happened on the latter too, regardless of the order. Also, the Click-through Rate (CTR) scores were expressed as percentages before any calculations. We take the absolute value as we are interested in the magnitude only. The results are presented in Table 8.2. To help interpret the table, the score listed in the cell Recency2-GeoRec corresponds to the difference in overlap between Recency2 and GeoRec obtained as the difference between the overlaps in performances of Recency2 and GeoRec when they ran in two identical online settings (which are represented by the logs of Recency2 and the logs of GeoRec).

The highest differences in overlap observed are between Recency2 and GeoRec and between GeoRec and RecencyRandom, each equal to **0.026**. Given that GeoRec and Recency are closely related algorithms, and Recency and Recency2 are identical, one would expect that the differences in overlap of GeoRec-Recency, and GeoRec-Recency2 should have been the same, and smaller than the difference in overlap of GeoRec-RecencyRandom. In an ideal evaluation environment, we would expect the difference in overlap to be equal to 0, because we would assume that the two settings under which the two algorithms run should affect the two algorithms in similar ways. Why do the two settings then affect the two algorithms in different ways? The positive scores of differences in overlap, we argue, are a results of the idiosyncrasies of the particular settings.

8.3. Recommender Systems Evaluations: Offline, Online, Time and A/A Test

In 2015, we participated in CLEF NewsREEL News Recommendations Evaluation (Task 1) by deploying four algorithms. In 2016, we reran four of our 2015 recommender systems without change. This allows us to compare the performance of the systems in 2015 and 2016. In 2016, we participated also in Task 2, which allows us to evaluate the recommender systems in a simulated environment and then compare the offline performance measurements

with the corresponding online performance measurements. This setup allows us to investigate recommender systems across offline, online, between years, and also to study the effect of non-algorithmic factors on performance. We present the results of these evaluations and comparisons along several dimensions and highlight similarities, differences and patterns or the lack thereof.

The four recommender systems we ran in 2015 and reran in 2016 are two instances of **Recency**, one instance of **GeoRec** and one instance of **RecencyRandom**. **Recency** keeps the 100 most recently viewed items for each publisher, and upon recommendation request, the most recently read (clicked) are recommended. **GeoRec** is the Recency recommender modified to diversify the recommendations by taking users' geographical areas and items' local and non-local categories. **RecencyRandom** recommends items randomly selected from the *100* most recently viewed items. For a detailed description of the algorithms, refer to [107].

In 2015, the recommender systems ran from 2015-04-12 to 2015-07-06, a total of 86 days. RecencyRandom started 12 days later in 2015. In 2016, the systems ran from 2016-02-22 to 2016-05-21, a total of 70 days. We present plots and tables for 2016: plots of daily performances, the cumulative performances as they progress, and the overall cumulative performances. The plot for the daily performance is presented in Figure 8.1. From the plot, we observe that there is a big variation in CTR between the recommender systems. The minimum is 0 for all of them. The maximums, however, vary a lot. The maximum CTR for Recency2 is 12.5%, for GeoRec 5.6%, for RecencyRandom 4.3%, and for Recency 4.2%. The maximum scores all occurred between the 18th day (2016-03-10) from the start of our participation and the 31st day (2016-04-08). The highest scores of Recency2, and GeoRec occurred on 2016-03-21, for RecencyRandom on 2016-03-20, and for GeoRec on 2016-03-10.

We do not exactly know why the performances showed increases on the dates between 2016-03-10 and 2016-04-08, and why all the highest scores for most of the three of the systems occurred on the two days. But, we observed a reduced number of recommendation requests between 2016-03-10 2016-04-06, when the systems showed increased performance scores. For some of the systems, no results are reported between the dates 2016-03-24 and 2016-04-05. If this reduction of recommendation requests was across all teams and systems, the lower number of recommendations and increased CTR might mean that users are more likely to click on recommendations when recommendations are offered sparsely. If this is the case, it might suggest an investigation into the relationship of the number of recommendations and user responses.

The plots of the performances as the systems progress on a daily basis are presented in Figure 8.2. This should be compared with the plots for 2015 which are in Figure 7.7. The cumulative number of requests, clicks and CTR scores of the systems in both years are presented in Table 8.3. The cumulative performances are all below 1, the average CTR score for news recommendation. The maximum performance differences between the systems are 0.16 in 2015 and 0.07 in 2016.

From the plots in Figure 8.2 and the cumulative performances in Table 8.3, we observe that the performances of the different systems vary. Are the performance variations between the different systems statistically significant? We look at statistical significance on a daily basis after the 14th day, which is considered the average time within which industry A/B

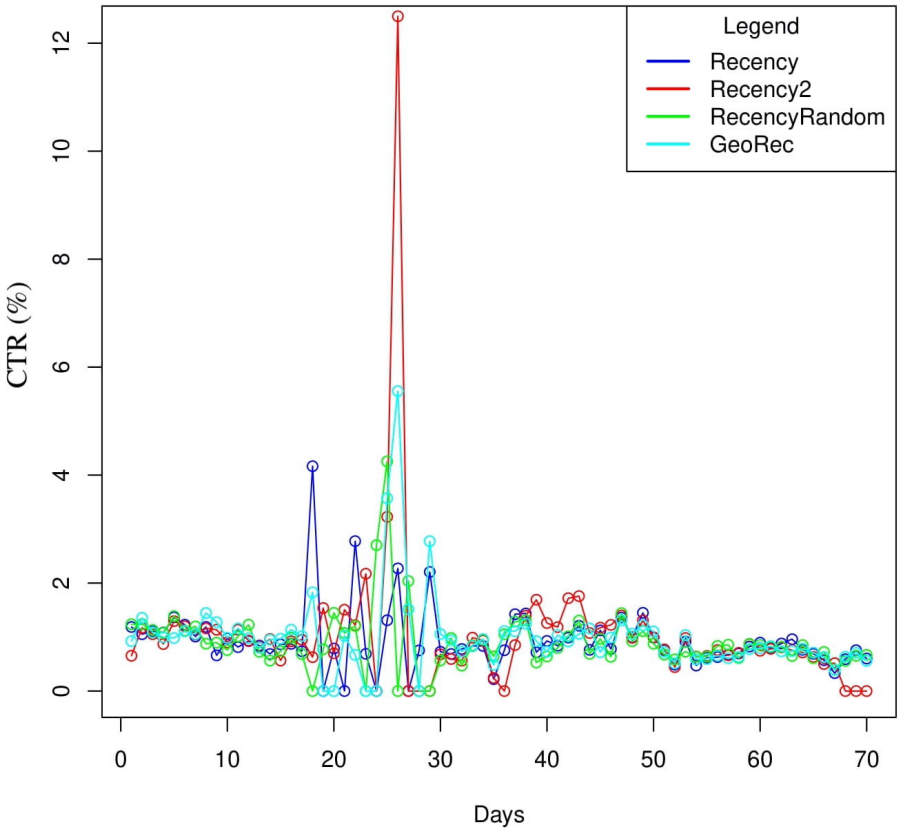


Figure 8.1: The daily CTR performances of the four online recommender systems in 2016. We note that there is a big difference between the days. Between 18th and 31th days, we observe unusual increases of the CTRs of all systems.

Table 8.3: Number of requests, number of clicks and CTR scores of four systems in 2015 and 2016.

	2015			2016		
Algorithms	Requests	Clicks	CTR(%)	Requests	Clicks	CTR(%)
Recency	90663	870	0.96	450332	3741	0.83
Recency2	88063	810	0.92	398162	3589	0.90
RecencyRandom	73969	596	0.80	438850	3623	0.83
GeoRec	88543	847	0.96	448819	3785	0.84

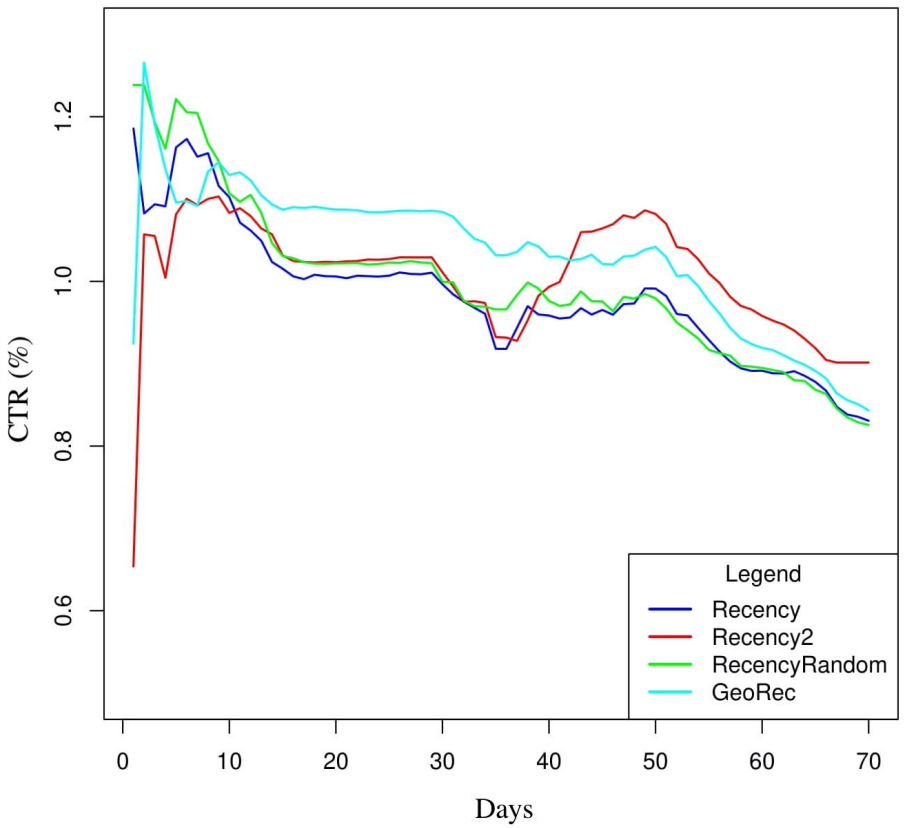


Figure 8.2: The cumulative CTR performances of the four online systems as they progress on a daily basis in 2016.

Table 8.4: Statistical significance counts and percentages over the baseline of Recency2. Level of significance =0.05.

Algorithms	2015		2016	
	#Sig Results	%	#Sig Results	%
Recency	2	2.7	27	47.4
GeoRec	25	34.3	8	14

Table 8.5: Statistical significance counts and percentages over the baseline of RecencyRandom. Level of significance =0.05.

Algorithms	2015		2016	
	# Sig Results	%	# Sig Results	%
Recency	20	27.4	0	0
GeoRec	41	56.2	5	8.8

tests are conducted.

We perform statistical significance tests on a daily basis to simulate the notion of an experimenter checking whether one system is better than the other at the end of every day. In testing for statistical significance on a daily basis, we seek an answer for this question: On how many days would an experimenter seeking to select a better system find out that one system is significantly different from the chosen baseline? We investigate this under two baselines: Recency2, and RecencyRandom. We present the actual number of days and the percentage of days on which significant performance differences were observed. The results for the baseline of Recency2 are shown in Table 8.4, and the results for the baseline of RecencyRandom are shown in Table 8.5.

We also looked into the error notifications received by our recommender systems in the 2016 period. The error types and counts for each system are presented in Table 8.6. RecencyRandom has the highest number of errors. There were only three types of errors. According to the ORP documentation[1], error code 408 is for connection timeout, error code 442 is for invalid format of recommendation response and error 455 is not described. RecencyRandom has the highest number of errors.

We aggregated the error messages by days. Out of the 70 days, Recency has errors on 16 days, Recency2 on 19 days, GeoRec on 24 days, and RecencyRandom on 51 days. All systems received a high number of error messages on some specific days, especially on 2016-04-07 and 2016-04-08. It is not very clear why large numbers of error messages are received on particular days, but we observed that most of the high-error days seem to be those that are at the beginning of the start of the systems, or at the beginning of a change of load (from low to high). Why does the RecencyRandom recommender have the highest number of invalidly formatted responses is not clear, because the format is exactly the same as for other systems. The only difference between RecencyRandom and the others is that it is slower and we suspect the higher number of errors must have to do with its slowness.

[1]http://orp.plista.com/documentation/download, accessed in July 2015

Table 8.6: Count of errors messages received by our recommender systems in 2016. Error code 408 is for connection timeout, error code 442 is for invalid format of recommendation response and Error 455 is not described. RecencyRandom has the highest number of errors.

.

Error Types	Recency	Recency2	RecencyRandom	GeoRec
408 (Connection Timeout)	40	118 40	14	159
442 (Invalid-Format)	1377	1390	26608	1360
455	281	217	348	252
Total	1698	1725	26970	1771

Online Evaluation

In the online evaluation, or Benchmark News Recommendations in a Living Lab (Task 1) as it is called in CLEF NewsREEL, we investigate recommender systems in two dimensions. One dimension is time where we compare and contrast the performances of our systems in 2015 and 2016. The second dimension is an A/A test where we attempt to study non-algorithmic effects on the performance of systems. Each of the dimensions are discussed in the following subsections.

Time Dimension: Performances in 2015 and 2016. The running of the recommender systems in 2015 and in 2016 gives us the opportunity to study systems from a time dimension. We compare the systems both in terms of their absolute and relative (in terms of their rankings) performance. To compare the absolute performances, we used the 2015 instance of the recommender systems as the baselines, and the corresponding 2016 instances as alternatives. The performances of the Recency and GeoRec instances of 2016 were significantly different from performances of the Recency and GeoRec instances of 2015 with a P-values of 1×10^{-4} and 9×10^{-4} respectively. The 2015 instances of Recency2 and RecencyRandom were not significantly different from their corresponding instances in 2016.

In 2015, Recency2 ranked third, but in 2016, it ranked first. In 2015, almost all systems started from lower CTR performances and slowly increased towards the end where they stabilized (see Figure 8.2) . In 2016, however, the performances of the systems reached their high at the beginning and then decreased steadily towards the end, except Recency2 which showed an increase after the first half of its deployment and then decreased (see Figure 8.2) . In 2016, the performances seemed to continue to decrease, and did not stabilize. unlike in 2015.

When we compare the daily significant performances in 2015 and 2016 (see Table 8.4 and Table 8.5), we observe that there is no consistency. In 2015, there were two days (2.7%) on which significant performance differences were observed between Recency and Recency2 while there were 25 days (34.3%) on which significant performance differences between GeoRec and Recency2. In 2016, Recency has shown 47.4% of the time significant performance, and GeoRec only 14%. When using RecencyRandom as a baseline, Recency has registered significant performance differences 27.4% of the time in 2015, and 0% in 2016. GeoRec has 56.2% in 2015 and 8.8% in 2016.

The performance of the systems defies any generalization in the dimension of time, under both baselines. The performance patterns vary a lot. The absolute and relative performances

in 2015 and 2016 vary from each other in the dimension of time. The implication of this is that one cannot rely on the absolute and relative rankings of recommender systems at one time for a similar job in another time. The systems have not been changed. The change, therefore, is coming from the setting where the systems are deployed. It is possible that the presentation of recommendation items by the publishers, the users and content of the news publishers might have undergone changes which can then affect the performances in the two years.

A/A Testing In both 2015 and 2016, two of our systems were instances of the same algorithm. The two instances were run from the same computer; the only differences between them were the port numbers by which they communicated with CLEF NewsREEL's ORP[2]. The purpose of running two instances of the same algorithm is to quantify the level of performance differences due to non-algorithmic causes. From the participant's perspective, performance variation between two instances of the same algorithm can be seen as pure randomness. The extent of performance differences between the instances can be seen as also happening between the performances of the other systems. We can consider that the performance difference due to the effectiveness of the algorithms is therefore the overall performance minus the maximum performance difference between the performances of the two instances.

The results of the two instances (Recency and Recency2) can be seen in Table 8.3, and in Figure 7.7 and Figure 8.2. The cumulative performances on the 86th day of the deployment in 2015 showed no significant difference. In 2016, however, Recency2 showed a significant performance over Recency with a P-value of 0.0005. Checking for statistical significance on a daily basis after the 14th day (see 8.4), in 2015, there were 2 days (2.7%) on which the two instances differed significantly. In 2016, however, that number of days was much higher, a total of 27 days (47.4%). This is interesting for two reasons: 1) the fact that two instances can end up having statistically significant performance differences, and 2) that these significant differences occurred. Many times in 2016, one instance achieved significantly better performances over the other instance.

8

Offline Evaluation

We present evaluations conducted offline, or in Benchmarking News Recommendations in a Simulated Environment (Task 2), as it is called in CLEF NewsREEL. Evaluation in Task 2 is different from other offline evaluation in that Task 2 is a simulation of the online systems. Usually, systems are selected on the basis of offline evaluation and deployed online. Other things such as complexity and latency being equal, there is this implicit assumption that the relative offline performances of systems holds online too. That is that if system one has performed better than system two in an offline evaluation, it is assumed that the same rank holds when the two algorithms are deployed online. In this section, we investigate whether this assumption holds by comparing the offline performances with the online performances of the algorithms in Task 1.

Task 2 of CLEF NewsREEL provides a reproducible environment for participants to evaluate their algorithms in a simulated environment that uses the user-item interaction

[2]http://orp.plista.com/, accessed in July 2015

Table 8.7: The performances of our algorithms in simulated evaluation (Task 2). For each system, there are the number of correct clicks (clicks), the number of requests, and the CTR (clicks1000/requests) and the number of invalid responses (Invalid). Results for publishers http://www.cio.de (13554), http://www.gulli.com (694), http://www.tagesspiegel.de (1677), sport1 (35774) and all are shown in the table.

Instance	Publisher	Click	Request	Invalid	CTR
Recency	13554	0	21504	0	0
	694	13	4337	0	2
	1677	69	46101	0	1
	35774	3489	518367	0	6
	All	3571	590309	0	6
RecencyRandom	13554	0	12798	1451	0
	694	3	4347	0	0
	1677	0	0	7695	0
	35774	2297	519559	0	4
	All	2300	536704	9146	3
GeoRec	13554	0	21504	0	0
	694	13	4337	0	2
	1677	69	46101	0	1
	35774	3445	518411	0	6
	All	3527	590353	0	5

dataset recorded from the online interactions [108]. In the simulated environment, a recommendation is successful if the user has viewed or clicked on the recommendations. This is different from Task 1 (online evaluation) where a recommendation is a success only if the recommendation is clicked. The performances of our algorithms in the simulated evaluation are presented in Table 8.7. The plots as they progress on a daily basis are presented in Figure 8.3. In this evaluation, Recency leads followed by GeoRec and then RecencyRandom. Using RecencyRandom as a baseline, there was no significant performance difference in both Recency and GeoRec. When we compare the ranking with the rankings of the systems in Task 2, there is no consistency. This, once again, shows that the relative offline performances of recommender systems do not show to hold online, much less the absolute performance.

From Table 8.7, we observe that only RecencyRandom has invalid responses. We also observed that RecencyRandom has higher error messages and lower performance in Task 1. To understand why, we looked at the response times of the systems under extreme load. The mean, min, max and standard deviations of the response times of the three systems are presented in Table 8.8. We observe that RecencyRandom has the slowest response time followed by GeoRec. We have also plotted the number of recommendations within 250 milliseconds in Figure 8.4. Here too, we observe that RecencyRandom has the slowest response time than the other systems. Given the operation of randomizing before selecting recommendation items, it is not surprising that it has the slowest response time. When we look at the publisher-level breakdown of the recommendation responses in Table 8.7, we see that RecencyRandom has invalid responses for two publishers, and for publisher Tagesspiegel (1677), all its recommendations are invalid. In the offline evaluation, invalid

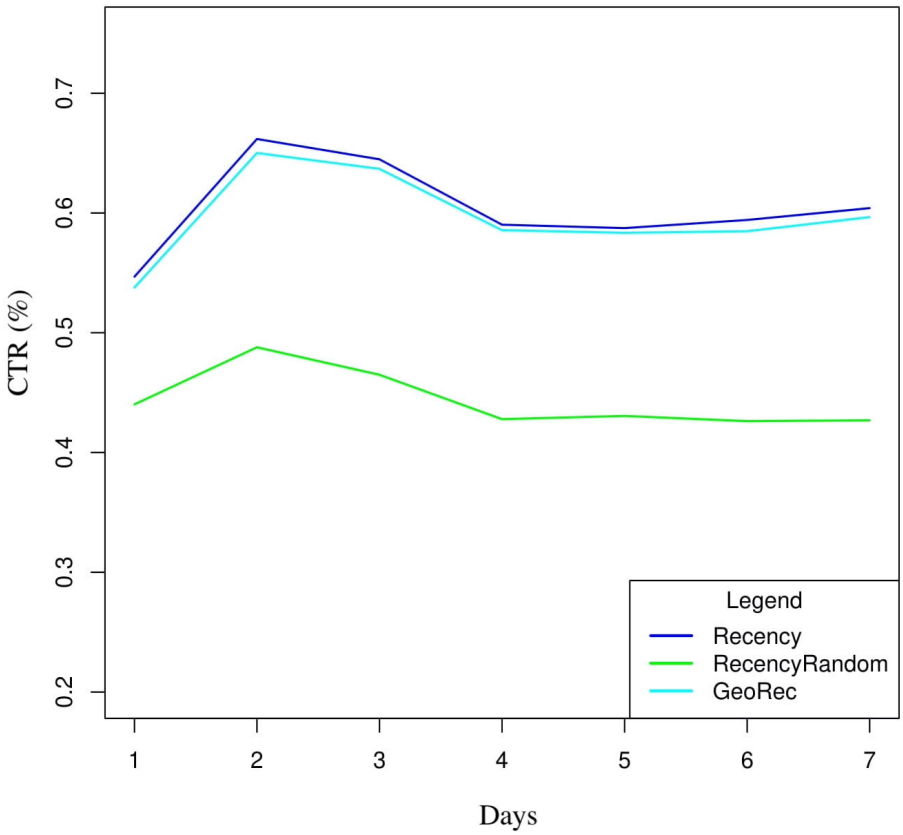


Figure 8.3: The CTR performances of the three offline systems as they progress on a daily basis.

Table 8.8: The response times in milliseconds of the recommender systems. RecencyRandom has the slowest response time.

	Mean	Min	Max	stDev
Recency	9.057	0.0	2530.0	41.619
RecencyRandom	83.868	1.0	5380.0	319.463
GeoRec	11.549	1.0	2320.0	56.570

response means that the response generates an exception during parsing. We looked into the recommendation responses of RecencyRandom, and compared the response for publisher 694 and 1677. The only difference we saw is that almost all items responses for publisher 1677 were empty. This, we assume, must have to do with the extreme load.

8.3.1. Discussion

Our systems are very similar to each other, in that they are slight modifications of each other. This means that it is expected that their performances would not vary much. We have analyzed the performance of our systems from the dimensions of online, offline, and time. We have also investigated the extent of performance difference due to non-algorithmic causes in online evaluation by running two instances of the same algorithms.

We have observed substantial variation along the four dimensions. The performance measurements in both absolute and relative sense varied significantly in 2015 and in 2016. More surprisingly, the two instances of the same algorithm did also vary significantly, both in the two years and within the same year. This is surprising and indicates how challenging it is to evaluate algorithms online. In the online evaluation, non-algorithmic and non-functional factors impact performance measurements. Non-algorithmic factors include variations in users and items that systems deal with, and the variations in recommendation requests. Non-functional factors include response times and network problems. The performance difference between the two instances of the same algorithms can be considered to reflect the impact of non-algorithmic and non-functional factors on performance. It can then be subtracted from the performances of online algorithms before they are compared with baselines and each other. This can be seen as a way of discounting the randomness in online system evaluation from affecting comparisons.

The implication of the lack of pattern in the performance of the systems across time and baselines, and more specially the performance differences between the two instances of the same algorithm highlights the challenge of comparing systems online on the basis of statistical significance tests alone. The results call for caution in the comparison of systems online where user-item dynamism, operational decision choices and non-functional factors all play roles in causing performance differences that are not due to the effectiveness of the algorithms.

Comparison With Other Teams

It is also useful to compare the performances of our systems with the performance of other systems from other teams that participated in 2016 CLEF NewsREEL's Task 1. Here, we consider the results for 28 April to 20 May, provided by CLEF NewsREEL to all the partic-

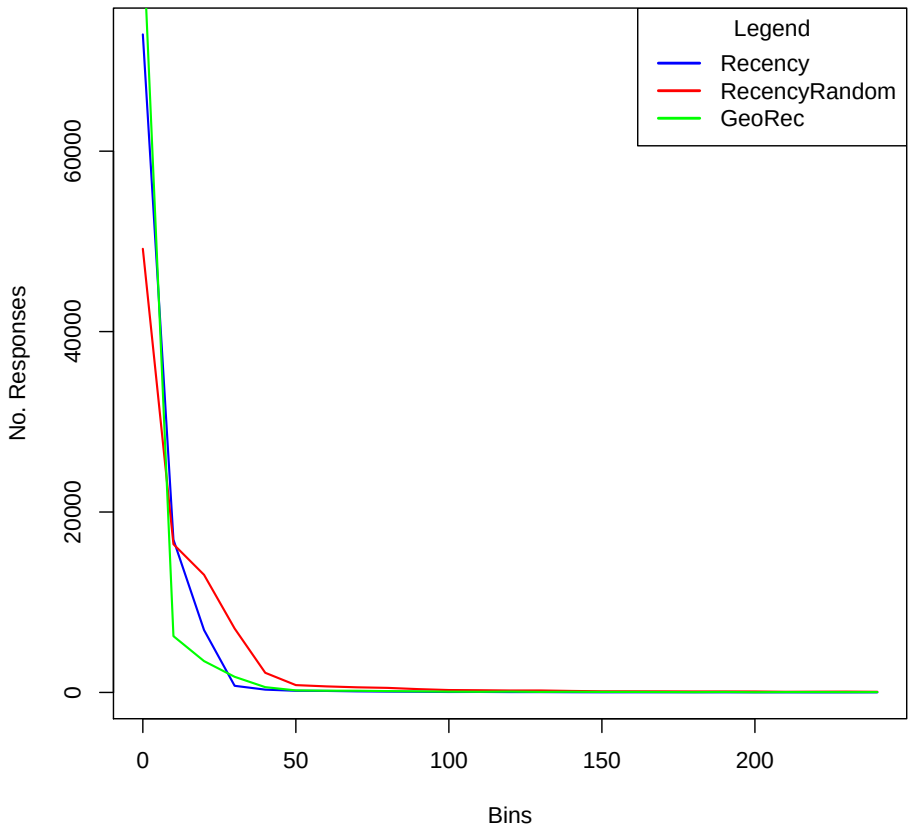


Figure 8.4: The number of recommendation responses against response times in milliseconds for the systems in Task 2.

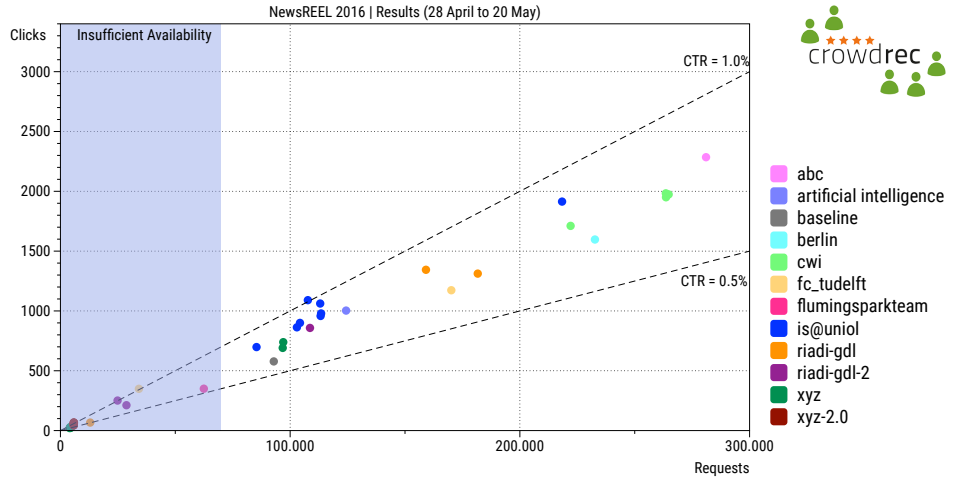


Figure 8.5: The rankings of the 2016 teams that participated in the CLEF NewsREEL challenge. CWI represents ours. The plot was provided by CLEF NewsREEL.

ipating teams. The plot of the team ranking as provided by CLEF NewsREEL is shown in Figure 8.5. We examined whether the performance of the best performing systems from the teams that are ranked above us (we are CWI) were significantly different from ours. Only the ABC's and Artificial Intelligence's systems were significantly different from Recency2 (our best performing system for 2016).

8.4. Discussion and Conclusion

The chapter started with investigating the causes of (random) performance differences in real-world recommender systems, and continued to examine recommender systems from multiple dimensions (time-wise, offline, online and A/A). In this section we discuss the results and conclude the chapter with an appraisal of the CLEF NewsREEL: News Recommendation Evaluation Lab platform, the platform where the experiments in this chapter and the previous one are conducted.

In Section 8.2, we investigated the performance differences in online algorithms. We employed several algorithms among which were two instances of the same algorithm. We classified and discussed the possible causes of performances differences between online-deployed algorithms and argued that even in the absence of obvious causes of performance differences such as operational biases and the selection of users and items observed in the experiment, performances can vary due to other artifacts in the data collected. These artifacts will also exist in offline datasets, but in the online setting, the researcher is much more susceptible to being misled by such artifacts, as it involves users and items and their dynamic interactions. We cannot claim that these artifacts are the sole reason for observed significant performance differences between two instances of the same algorithm; and forming an important confounding factor when comparing any two algorithms in general. We may however conclude that we have to take into account these random biases that can only

be smoothed out over a sufficiently long evaluation period.

Our results suggest that we should be reluctant in adopting (even statistically significant) improvements as indicative of real performance differences when the evaluation involves real world settings, users and items. We have proposed a new method to quantify the effect of randomness in the evaluation by zooming in on the differences in overlap of the results obtained from two competing algorithms that are tested on two settings simultaneously.

In Section 8.3, we set out to investigate the performance of recommender system algorithms online, offline, and in two separate periods. The recommender systems' performances in different dimensions indicate that there is no consistency. The offline performances were not predictive of the online performances in both absolute and relative sense. Also, the performance measurements of the systems in 2015 were not predictive of those in 2016, both in relative and absolute sense. Our systems are slight variations of the same algorithm, and yet the performances varied in all dimensions. We conclude that we should be cautious in interpreting the results of performance differences, especially considering the performance differences between the two instances of the same algorithm.

This chapter would be incomplete without an appraisal of the CLEF NewsREEL, the platform under which all the experiments in this Part were conducted. In particular, we discuss opportunities, validity, and fairness.

8.4.1. Opportunities

CLEF NewsREEL has provided a unique opportunity for researchers working on recommender systems. It has enabled researchers to test their algorithms in a real-world setting with real users and items. In addition, participants competed with one another. Thus, they get feedback on how their algorithms compare with competitors' algorithms. Further, participants have gained access to a large number of log files comprising interactions between users and items. They can conduct offline experiments with these data thus optimizing their system before deploying them. Researchers hardly have access to such conditions otherwise, making CLEF NewsREEL a unique form of benchmarking.

8.4.2. Validity and fairness

Participants seek to compare their algorithms with competing algorithms. They need to know how valid comparisons are to estimate how well their systems will perform in the future. Determining validity represents a challenging task. Unlike the operators of recommender systems, participants only perceive parts of the environment. Various effects can potentially bias observed performance.

We distinguish operational and random biases, the latter resulting from random effects such as the dynamics in user and item collections. Operational bias refers to the result of operational choices of the evaluation framework, including those that lead to favoring some participants' systems over others or delegating a disproportional number of requests from specific publishers to a few systems only. The latter in particular would skew results, as items originating from specific publishers have been found to receive a stronger user response.

Fairness of the competition is closely related to the validity of findings, especially when considering operational biases. A (limited) level of random bias due to dynamic fluctuations in user and item collections is to be expected, but it would be very useful to be able to quantify its influence. In the absence of biases, we would expect to observe similar performance

Table 8.9: Data collected by running two instances of the Recency recommender in the 2015 and 2016 editions of NewsREEL.

	2015			2016		
Algorithms	Requests	Clicks	CTR (%)	Requests	Clicks	CTR (%)
Instance1	90663	870	0.96	450332	3741	0.83
Instance2	88063	810	0.92	398162	3589	0.90

of identical systems over sufficiently long periods. Therefore, we have applied a method of evaluation that is best described as A/A testing; unlike in the usual A/B testing, A/A testing subjects the users to different instances of the same algorithm. The instances were run on the same computer and the same environment; only the port numbers they used to interact with Plista were different. With this setup, we do not expect the ORP to treat the two algorithms differently since their behavior should be identical. Since the same algorithm was used to generate the recommendations, we attribute differences in the responses by users to those recommendations to bias or randomness, and we analyze those differences to quantify their effect.

Experiment

As participants, we experimented to estimate operational and random biases in CLEF NewsREEL. We set up two instances of the same recommendation algorithm, implementing an A/A testing procedure. We implemented a recency-driven recommender, which keeps the 100 most recently viewed items and suggests the five or six most recent upon request. Random biases may cause performance variations on a daily level. In the absence of operational biases, we may expect these performance measures to converge in the long term. Both instances of the recency recommender have run in NewsREEL's editions in 2015 and 2016. In 2015, the two instances ran from 2015/04/12 to 2015/07/06, a total of 86 days. In 2016, both instances ran from 2016/02/22 to 2016/05/21, a total of 70 days. We considered only the recommendation requests and clicks of days on which the two instances of our algorithms ran simultaneously. Table 8.9 presents requests, clicks, and the CTR for both periods. The observed difference in CTR is small, 0.04% in 2015 and 0.07% in 2016, based on which we conclude that the evaluation does not show evidence of an operational bias. On the other hand, we notice a marginal level of random bias. Figure 8.6 shows the average CTR as a function of the number of days, for the year 2015 and Figure 8.7 for the year 2016. Initially, we observed fairly high levels of variance between both instances in 2015. Over time, the variance levels off and both instances of the algorithm approach a common level of $\approx 8.5 \times 10^{-1}$ %. In 2016, we observed the opposite trend in that the algorithms perform more similarly and diverge towards the end.

Log Analysis

We noticed that A/A testing with two instances of the same algorithm results in performance variations that, in 2015, smoothed out when observed over a sufficiently long period, but in 2016 showed divergence towards the end. We analyzed our log files from 2015 to identify two hypotheses to explain these variations. First, operational bias might induce an unfair setting, in which some instances naturally perform better than others. Alternatively, random

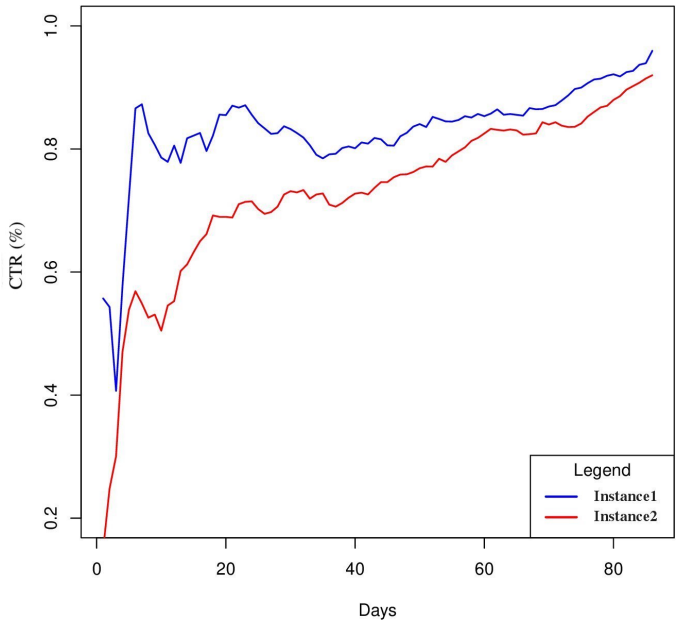


Figure 8.6: The cumulative CTR performances of the two instances as they progress on a daily basis in 2015.

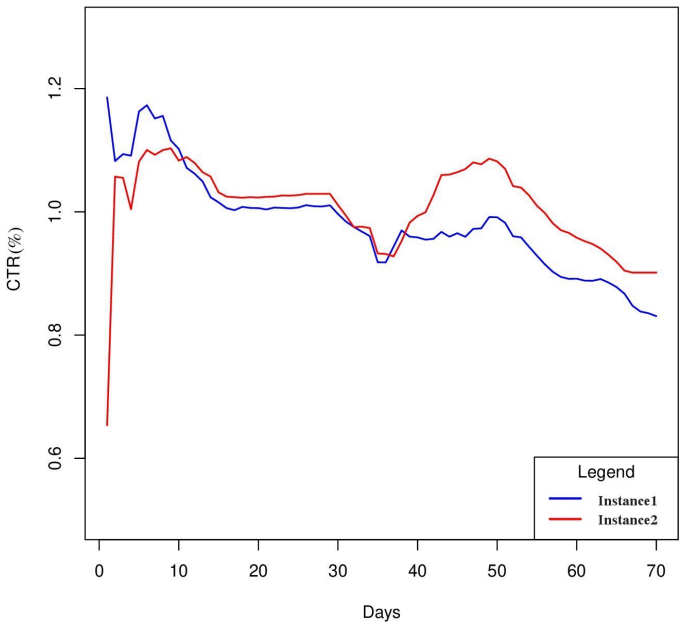


Figure 8.7: The cumulative CTR performances of the two instances as they progress on a daily basis in 2016.

bias due to the selection of users and items presented to each recommender may explain the performance variation observed.

Analyzing Recommendation Requests by Publisher: We look into the distribution of requests across publishers. In a fair competition, each participant will be subject to a similar distribution across publishers. We aggregated all requests on a publisher level for both instances. Subsequently, we computed the Jensen-Shannon Divergence (JSD) metric to quantify the differences between both distributions. We obtained a divergence score of approximately 3×10^{-3}, indicating that both instances received similar distributions of requests. At the level of a publisher, we conclude that we did not find a noticeable bias that would be attributed to operational design choices in the evaluation framework.

Analyzing Recommendation Requests and Responses at Item and User Levels: We investigate the overlap between the sets of users and items processed by both instances, by measuring their Jaccard similarity; high overlap would signal the absence of random biases. Comparison of the sets of items produced a Jaccard similarity of 0.318 whereas the sets of users resulted in a score of 0.22. Given the low overlap between users and items presented in both instances, we conjecture that the chance to observe the same user on both systems is relatively low (which can be explained by the limited number of events in each session). We note that the overlap is impacted by the fact that there are tens of other systems running simultaneously. The observed overlap is consistent with the conclusion that user and item variation arises due to natural dynamics.

8.4.3. Concluding Remarks

Our focus has been on understanding the perspective that is accessible to the participants on whether or not the NewsREEL evaluation treats all participating algorithms fairly. We reported on the results of A/A testing conducted to estimate the level of variance in CTR for identical algorithms. We hypothesized that random effects or operational design choices could cause varying performances. We observed varying trends, in 2015 and 2016, in the cumulative performances of the two instances. In 2015, the variance diminished over time, but in 2016 the variance emerged later. We analyzed the logs of our participating systems to determine which kind of effect produced the variance. We found that requests were distributed equally across publishers for both instances. Based on this observation we were able to conclude, from the participant's perspective, that operational design choices are unlikely to have caused the variance. Instead, we observed that collections of users and items differed between both instances.

From the participant's perspective and the current setup, it is possible to conduct a partial investigation into possible operational biases and have a reasonable estimate of the impact of those causes on the performance of a participating system. We conclude that participants do have the means to assure themselves of NewsREEL's fairness using only information available from the participant's perspective. We note, however, that an exhaustive investigation of all possible operational biases is either too complicated and/or impossible from the participant's perspective. For example, operational biases could be implemented at the level of pairing logged-in and logged-off users to different teams or participant systems, pairing some item categories to some participants or systems, and disfavoring one system

based on response and other network factors. The possibility to explore some of the biases is somewhat hampered by the fact that participants do not receive direct information on whether their recommendations are clicked. It is possible to extract a system's recommendation clicks from the logs, but it requires complicated implementation and is also subject to error. The error is in turn dependent on how the participant chooses to implement the mapping of recommendations to clicks.

9

New Developments in News Recommendation

9.1. Introduction

Part II investigated news recommendation, with a particular focus on evaluation. All the works are based on our participation in CLEF NEWSREEL News Recommendation Evaluation Lab. We started participation in 2014 and continued our participation in 2015 and 2016. We use the datasets obtained from our 2014 participation to investigate the role of geography in news consumption. The investigation is a descriptive study of the relationship between the geographical focus of a news item and the geographical location of the users. We observed that geographical information plays a role in users' decision of whether to read news items or not. We followed it with online deployments of algorithms, where we also incorporated geographical information in some of them. The incorporation did not readily translate into a noticeable performance improvement. In addition to this, we also attempted to quantify random fluctuations in the performance difference of a live recommender system.

After that, we focused on news evaluation, investigating it from several angles. We conducted A/A tests, offline evaluations, online evaluations, and comparisons of algorithm performances across years. Our findings show that recommender system performances fluctuate across all of these dimensions. Offline performance does not hold online, both in absolute and relative sense. The A/A test shows the challenge of evaluating a news recommender system in a dynamic environment where there is a constant churn of items and users and user's evolving interests. Algorithm performances, both absolute and relative, in one year, do not hold in the next year. Based on these findings and observations, we recommended that extra care, in addition to statistical significance tests, be taken with reported improved performances. Finally, we appraised the CLEF NEWSREEL News Recommendation Evaluation Lab from our perspective and forwarded recommendations for improvement.

In this brief chapter, we review developments in News Recommendations in general with a focus on those that have some bearing on the experiments and findings reported in Part II.

9.2. Review

CLEF NewsREEL continued in 2017 with an increased volume of messages and the addition of new news portals [109], but the tasks remained the same. In 2018, CLEF NewsREEL continued as NewsREEL Multimedia as part of MediaEval [12], but the task was different. The task of NewsREEL Multimedia was, given text snippets about news items, features of the associated images, and user-item interaction features, to learn a model to predict an item's popularity. Popularity was determined by the number of visits an item received.

Approaches to CLEF NewsREEL tasks in the years before 2017 were based mainly on popularity-recency of items [110]. In 2017, we observed new approaches such as graph-based recommender algorithms, hybrid approaches, contextual bandit and an approach using a neural network architecture.

Information access systems are evaluated in four major ways: offline with static test collections, small-scale user studies, user simulations or online evaluation settings [111]. Offline evaluation, which is the most popular one, is favored mainly for reproducibility. But it may not reflect user satisfaction [112] and standardized datasets can have drawbacks because algorithms can be fine-tuned to the datasets [111]. These limitations of offline evaluation have led to the adoption of online evaluations which use online information systems and actual users to evaluate information systems. Online evaluations can be either A/B testing (the most common) where some users are exposed to system A and a disjoint set of other users are exposed to system B, or interleaved comparisons where items from two or more recommender systems are combined (interleaved) [111].

Recommender systems in CLEF NewsREEL 2015, 2016 and 2017 were evaluated in the online environment and the simulated environment. In both cases, participants' systems were compared using CTR scores. NewsREEL 2018 was very different from the other tasks, notably because it required that systems predict an item's popularity which, by definition, disallows personalization. This also means the use of different performance metrics. Specifically, NewsREEL 2018 used precision@N which measures how precisely systems identify the most popular N items, and average precision, which is the mean of the top N precision scores.

Recently, Neural Network approaches to news recommendation are being increasingly adopted [113–119], but all of them use offline datasets and evaluation. A recent work [72] investigated whether the increasing application of neural network approaches to the task of top-N recommendation is resulting in progress. After selecting many works from main conference proceedings and journals, they tried to reproduce and compare them with what they call "weak baselines". They draw two key conclusions: 1) only one-third of the works were reproducible with reasonable effort and 2) almost all of them were outperformed by the baselines.

Regarding evaluation metrics, CTR remains dominant, but there is an increasing attempt to devise other metrics. For example, there are attempts to measure serendipity, novelty, diversity, coverage and others [18]. There is also interest in measuring recommender system performances using dwell time or the likelihood that users will return [111], and some of the recent works have attempted to measure the likelihood of user return [116], albeit only in offline evaluation. Other studies use other metrics such as AUC, MRR, nDCG [114].

A recent study argues that recommender systems should be evaluated according to a chosen normative stance and proposes four evaluation metrics conforming with four demo-

cratic models [23]. Another study investigates the impact of target sampling (the choice of the set of item-user pairs) in the evaluations of recommender systems [120].

The experiments in this Part of the thesis pursue several goals. The first goal is to investigate and quantify random performance differences between online news recommender systems. These are differences not attributable to the algorithms themselves but result from the chance and the intricacies of the system. The second goal is to examine news recommender systems evaluations along several dimensions: offline evaluation, online evaluation, across time, and the effect of non-algorithmic factors on the performance of an online recommender system by using an A/A test. The third and final goal is to appraise the CLEF NewsREEL News Recommendation Evaluation initiative—the platform where the experiments were conducted—from a participant's (our) perspective.

Abstracting away from the specific CLEF NewsREEL initiative, our experiments investigate 1) whether there are random performance differences between online recommender systems that are not attributable to the works of the algorithms themselves 2) whether the absolute or relative offline performance of algorithms hold online too 3) whether evaluation patterns of online systems hold across time 4) whether an online A/A test shows significant performance difference. The research goals that we pursued and the experiments we conducted are independent of the CLEF NewsREEL initiative. They are also independent of the particular algorithms in the sense that they do not assume any particular types of news recommender systems.

The particular performance results reported are, however, dependent on the CLEF NewsREEL datasets and settings. For example, we report that two identical recommender systems have shown a significant performance difference. We also quantified what we called random performance differences between recommender systems using a method we called difference in overlap, and reported that the difference is significant. We also report the performance differences across time and the performance differences of A/A testing. Whether these conclusions hold in another online setting using a different type of algorithm (for example, neural net) is not yet replicated or reproduced.

The ACM task force on reproducibility states "a scientific result is not fully established until it has been independently reproduced" [121]. According to the PREMAID—Platform, Research Goal, Implementation, Method, Actor, Data— a model of reproducibility, an experiment is reproduced by changing—priming— some of the components of the experiment [121, 122]. The components of our experiments are specified. The research goals and the methods are simple and well-described. Data is CLEF NewsREEL (2015 and 2016) which is well described in many working notes and overview papers, and the platform is partly the ORP (also well-described, but currently not available) and our computers from which we run the systems. Implementations of our algorithms are found in Github[1]. We believe the components are well described for reproducibility. Our experiments have, however, not been reproduced by others (yet).

9.3. Conclusion

While there are generally developments in the area of News Recommendation, there are no works directly replicating or disputing our experiments and findings. The reported results

[1]https://github.com/gebre/, accessed on 6 December 2024

and conclusions await further experiments by priming one or more of their components.

III

Measuring Personalization

10

Pull–push: A Measure of Over- or Underpersonalization in Recommendation

A recommender system imposes differences between users by presenting to them different recommendation lists, which they respond to, resulting in different "reaction" lists, which are the lists of items they have chosen to act upon (for example, by clicking on). Comparison of the differences in the recommendation and reaction lists can indicate different user to use relatitionships. Users can approve the imposed difference, end up narrowing the difference between them (pulling each other closer) by consuming more of the items in common or enlarge the difference between them (pushing each other further apart) by consuming the items not in common. When users do not approve the differences, they are either in a push state (implicitly disapproving underpersonalization) or in a pull state (implicitly disapproving overpersonalization). We offer the pull-push metric to quantify the magnitude of pull or push—measures of disapproval by the users of, respectively, overpersonalization and underpersonalization. Application on simulated datasets shows that users can push each other away up to having disjoint sets of items or pull each other closer up to having identical sets. On real-world datasets, we find that the particular recommender system was underpersonalizing its recommendations. We show how the pull-push metric can be merged with another metric of personalization to come up with a measure of the potential for improvement in a recommender system, and discuss its relationship to popularity bias.

10.1. Introduction

Recommender systems, e-commerce sites, social media platforms, and search engines use personalization to tailor recommendations to their users [123, 124]. It is a core part of content consumption and companies' revenue [125].

A recommender system's (RS's) personalized recommendation is meant to help users find information of their interest by filtering items by their relevance. This reduces users'

information overload and information seeking efforts, saves them time and improves their decision quality [8]. For providers, a RS is useful to keep users engaged and retain a variety of audiences, thereby to increase revenue. Recommendation providers and users have, therefore, a common interest in meeting users' needs and preferences using personalized recommendations.

A RS's personalization can be measured for different ends and from different perspectives. For instance, Aniko Hannak et al. [124] measure personalization as the variation from the presentation of search results in exact order for each query for each user. In doing so, this measure totally disregards the user differences. It is evident, however, that users have differences in information needs and interests, even when they use the same search query [126]. They are also able to single their preferences out from a mixed presentation [127–130]. The challenge for a personalized RS, then, is to be able to capture the natural and healthy differences between users without imposing new and unnecessary differences. This suggests the need for a precarious balance where users' differences in interests and tastes are satisfied, where they are neither "overloaded" with content they are not interested in, nor are they served with content more differentiated than necessary.

How do we then measure RS's personalization success in meeting the users' differences? Personalized recommendation can be conceived of as a two-stage process: the generation of the recommendation list and a subsequent ranking of the recommendation list. By choosing different recommendation lists to users, a recommender system imposes a difference between the users. The users react to the differentiated recommendations, for example by clicking. Implicit in their reactions, users approve the imposed difference when there is a proportional difference in their reactions to the differentiated recommendations. Likewise, they may also implicitly show a wish to pull (get closer to each other) by consuming more of the items in common between the two lists, or a desire to push each other (drift apart) by consuming more of the items that are distinct. The differentiated recommendations and the subsequent reactions by the users give us two groups of content differentiation. Comparison of the two groups of differentiation allows us to measure the degree of pull or push that a RS imposed difference causes.

We introduce a user-centric metric called pull-push, to quantify the discrepancy between the two groups of content differentiation. Pull-Push compares the degree of differentiation in recommendations with the degree of differentiation in the resulting user reactions. The metric's score can indicate three states: balanced, pull or push. A pull state indicates the users' tendency to come together (consume common items) despite the imposed differences in the recommendations. A push state, on the other hand, indicates the tendency of users to drift apart more than that which is imposed by the RS. In this situation, the RS should personalize more. A balanced state indicates a congruence between recommendations and user interests/preferences. A pull or a push state shows a degree of disapproval of the recommendations by the users. A pull score is a measure of over-personalization, and a push score is a measure of under-personalization. Ideally, we would want a deployed RS to be in a balanced state.

The contributions of our work are: 1) a novel user-centric conceptualization of a recommender system's content differentiation success 2) a generic, and versatile user-centric metric for quantifying the gap between personalization and user interests 3) applications of the metric on simulated and real-world datasets and 4) discussion of the metric in relation to

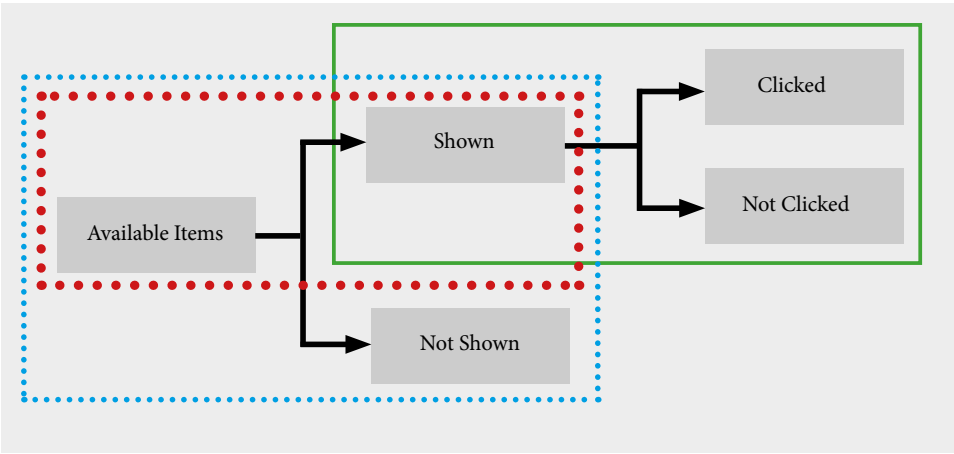


Figure 10.1: The recommendation flowchart from available items to clicks. Available items are either shown (recommended) or not shown. Shown items are either clicked or not clicked.

other metrics on personalization, to popularity bias, and to normative standards. The rest of the paper is organized as follows. In Section 10.2, we present background and related work. In Section 10.3, we discuss our proposed method, followed by Section 10.4 where we experiment with the application of the proposed method on simulated and real-world datasets. We discuss the method and the results in a broader context in Section 10.5 and finish with a conclusion in Section 10.6.

10.2. Background and Related Work

Here, we review relevant metrics on personalization. To help us explain them better, we use the recommendation flowchart shown in Figure 10.1. The flowchart shows a recommendation flow from item selection from an item pool, to impression and to interaction. This simplified flow applies to both query-based or query-less personalization. Available items can, for instance, be the daily news items for a news recommendation platform, or not-yet-personalized first-page search results. Different recommender system evaluation metrics address different aims and stages in this recommendation flowchart.

The area of interest for the metric proposed by Aniko Hannak, et al. [124] that quantifies personalization as the variation from exact presentation is the fine-dotted rectangle (Available Items + Shown + Not Shown), in Figure 10.1. In this point of view, a score is produced by comparing the personalized recommendations against each other, or against the available items. Accuracy-oriented measures such as RMSE and MAP, and engagement-oriented measures such as CTR and dwell time (see [131]) target the area surrounded by the solid-rectangle (Shown + Clicked + Not Clicked).

Works by Nguyen, et al. [25] and by Teevan et al. [126] target personalization at the level of the production of recommendation lists and at the level of ranking those lists respectively. Nguyen, et al. examined the effect of an item-item recommender system on the diversity of recommended and consumed items [25]. Similar to our approach, this chapter compares

recommendation lists and the resulting reaction lists, but while the pull-push metric aims to measure how successful the content differentiation is, the aim of the former is only to uncover personalization's impact on content diversity. Nguyen, et al.'s metric targets the boxes, in Figure 10.1, of Shown and Clicked directly, and the box of Not Clicked indirectly.

Teevan et al.'s "potential for personalization" [126] concerns itself with the ranking of the recommendation (search) lists. Using normalized DCG[132] as a measure of the quality of the ranking of a recommendation list, they define the difference between the ideal ranking score for an individual and that for a group as a potential for personalization. They concern themselves only with the ranking of the search (recommendation) list, ignoring the personalization done in the selection of the recommendation list. This chapter concerns itself with the boxes, in Figure 10.1, of Clicked and Not Clicked.

While a lot of measures are employed to assess RS's, methods to quantify RS's success at the content differentiation at the level of the generation of the recommendation list itself are under-addressed. To address this, we first conceive of personalization as containing two stages, namely content differentiation to generate the recommendation list and a subsequent ranking of the recommendation list. We then offer a novel, user-centric metric, which we call pull-push, that measures the gap between the degree of the RS's imposed content differentiation and the resulting reaction lists that are produced by the users given the differentiated recommendations. The gap between the difference in the recommendation lists and the difference in the reaction lists is a degree of the users' approval or disapproval of the imposed differentiation. A disapproval indicates either a disapproval of under-personalization or over-personalization.

Our pull-push metric differs from "the potential for personalization" in that it deals with the content differentiation and production of recommendation lists, as opposed to the ranking of a recommendation list. In our case, the potential for personalization, if there is one, is a function of the difference in the items of recommendation lists; in the Teevan et al.'s case, it is a function of the ranking of the recommendation lists. Our measure and Teevan et al.'s are complimentary, covering both the generation of recommendation lists and the subsequent ranking of the recommendation lists, and we will show how they can be combined to produce a score for potential for improvement later in this work. The entire area (Available Items + Shown + Not Shown + Clicked + Not Clicked) of Figure 10.1 can be targeted by the pull-push metric, as shown later when we discuss the application of the metric.

10.3. Method

Starting out from a balanced level of personalization in Figure 10.2, one can go either in the direction of less personalization or more personalization.

A good recommender system must strive to find the middle ground, the right level of personalization. The pull-push metric measures a recommender system's content differentiation level with respect to the balanced level. The following two concepts underpin our metric.

Differentiation Through Pair-wise Difference

Personalization imposes differences between recommendation lists for different users on the basis of a certain user model. This content differentiation happens in an environment that is in a state of flux, where both items and user interests are dynamic and

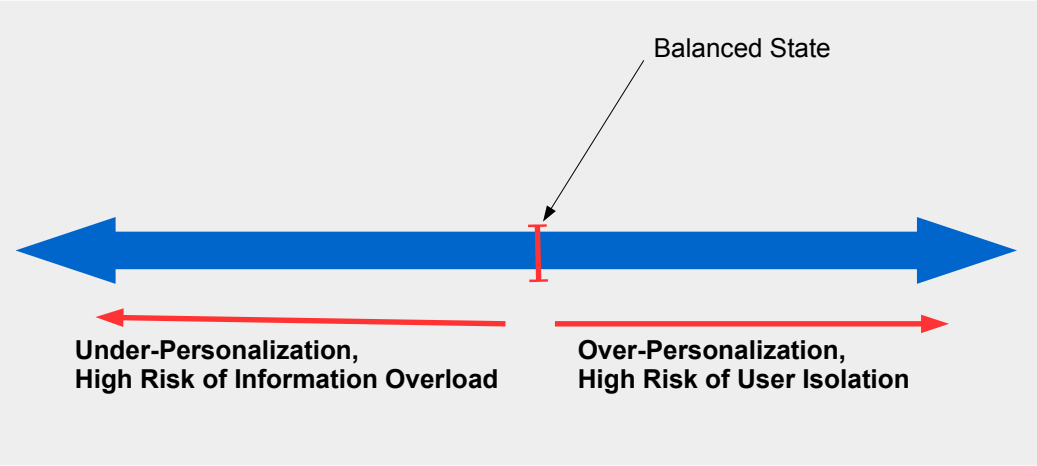


Figure 10.2: Under-personalization - Over-personalization. Starting from a balanced position, one can either go in the direction of under-personalization risking information overload, or in the direction of over-personalization risking user isolation.

evolving. We, therefore, cannot construct a stable "reference frame" for the computation of the success of the production of differentiated recommendations. To overcome this, we conceive of content differentiation as a function of the pairwise differences in the user recommendation lists and the resulting reaction lists. This conceptualization abstracts away from the actual recommendations and the resulting reactions, requiring only the preservation of the proportionality of similarities/differences in the recommendation lists and in the reaction lists.

User-centricity

From the user-centric perspective, it is important to satisfy the natural differences in interests and preferences. When the proportion of difference a recommender system imposes during recommendation is approved by the users in their reactions/consumption, we consider the recommender system successful in meeting the actual user differences. When that is not the case, the users are either in a state of pull or push, expressing disapproval, implicitly, by their selective behaviors.

User clicks on recommendations or lack thereof are reactions to the personalized recommendation list. It is fair to assume that an alteration in the recommended items would result in the alteration in the clicked items. This coupled nature of recommendations and subsequent user reactions means that user reactions are at best a tentative representation of the user's actual interest. In this work, we abstract away from the actual recommendations and the resulting reactions, and view a RS's differentiation success as the proportionality of the differentiation in the recommendations and the differentiation in the resulting user reactions. Proportionality is easier to observe and likely more stable, as it is independent of the particular recommendation lists and the resulting reaction lists.

Let X and Y represent the two recommendation lists presented to two users. $X \cup Y$ is the union of the recommendation lists, and $X \cap Y$ is the set of the shared recommendations (the

intersection). The proportion of the intersection of the items to the union of the items (see Equation 10.1) is the magnitude of similarity that the RS has imposed between the users. From this, we define σ_{rec} (see Equation 10.2), which is the magnitude of difference that the RS imposed between the pair of users.

$$J(X,Y) = \frac{|X \cap Y|}{|X \cup Y|} \tag{10.1}$$

$$\sigma_{rec}(X,Y) = 1 - J(X,Y) = 1 - \frac{|X \cap Y|}{|X \cup Y|} \tag{10.2}$$

The pair of users will react to the differentiated recommendation lists resulting in corresponding reaction lists X' and Y'. Like in the recommendations, we can obtain the set of shared items (intersection) and the set of the union of the items. The proportion of the shared items to the union of items is the magnitude of similarity between the two users according to the users themselves, given the differentiated recommendations. Using this, we also define σ_{react} as the magnitude of difference observed between the two users as in Equation 10.3.

$$\sigma_{react_{(X',Y')}} = 1 - J(X',Y') = 1 - \frac{X' \cap Y'}{X' \cup Y'} \tag{10.3}$$

If the magnitude of the difference in the reactions is the same as the magnitude of the difference that was imposed in the recommendations, that is $\sigma_{rec} = \sigma_{react}$, then the users seem to approve, implicitly, the RS's differentiation. We define this condition as the balanced state. We consider this condition ideal, assuming that satisfying the interests of both the recommendation provider and the users is the goal. The differences in the recommendations and in the reactions can also differ. The following are all the possible scenarios.

Balanced State $\sigma_{rec} = \sigma_{react}$
: This is the situation where the proportion of shared items to the union of items between the pair of users in the recommendation holds in the reactions of the users to the recommendations. Alternatively, this is the state where the distance in the recommendation lists (σ_{rec}) and the distance in the resulting user reaction lists (σ_{react}) are the same.

Push $\sigma_{rec} < \sigma_{react}$
: This happens when the proportion of shared to the union of items in the recommendations is greater than the one in the reactions. Alternatively, this is the state when the distance in the recommendations (σ_{rec}) is less than the distance in the resulting user reactions (σ_{react}). This happens when users diverge from each other—hence pushing each other away—by consuming proportionally larger number of items not in common and less of the shared items. This signals that the RS's differentiation of the recommendation lists is under-personalized. The bigger the magnitude of the difference between the distances, the larger the under-personalization according to the users.

Pull $\sigma_{rec} > \sigma_{react}$
: It is the opposite of push, and happens when the proportion of shared items to the

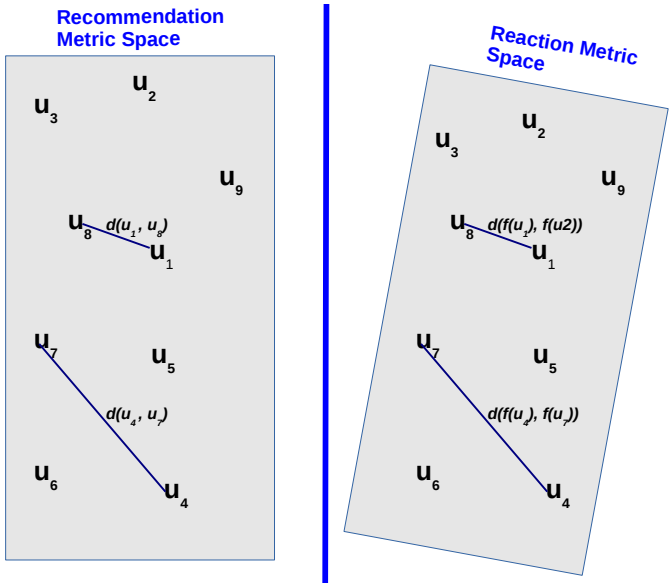


Figure 10.3: The mapping of the recommendation space to the reaction space. For a balanced content differentiation, the distances between the users in the recommendation space must be preserved in the mapping to the reaction space. That means, for example, $d(u_1, u_8) = d(f(u_1), f(u_8))$.

union of items in the recommendations is smaller than the one in the reactions. In terms of distance, this is the state when the distance imposed in the recommendations (σ_{rec}) is greater than the distance observed in the resulting reactions (σ_{react}). This happens when users come closer to each other—hence pulling each other—by consuming proportionally a larger number of items in common, and less of the items not in common. This signals that the RS's differentiation in the recommendation lists is over-personalized beyond the users want it to be between them.

We can now delve into a more detailed explanation of the pull-push metric. Let recommendation space be the set of users in the recommendations with a distance metric on the elements, and reaction space the set of users in the reactions to the recommendations (e.g. user clicks) with the same distance metric on them. We view the users in the recommendation space, and in the reaction space as metric spaces that are related to each other by a mapping, as shown in Figure 10.3. For a recommender system to be said in the balanced state, the mapping from the recommendation metric space to the reaction metric space must preserve the distance between users. Mathematically, the mapping from the recommendation metric space to the reaction metric space must be isometric. For our case, it means the distance between each pair of users u_i and u_j must remain the same before and after the mapping, i.e. Equation 10.4 must hold. It is this state of isometry that we consider a recommender system's balanced personalization. A good recommender system should be able to impose differences in the recommendation lists that result in proportional, distance-preserving reactions. Deviation from the balanced personalization is then considered a measure of disapproval of the degree of personalized differentiation.

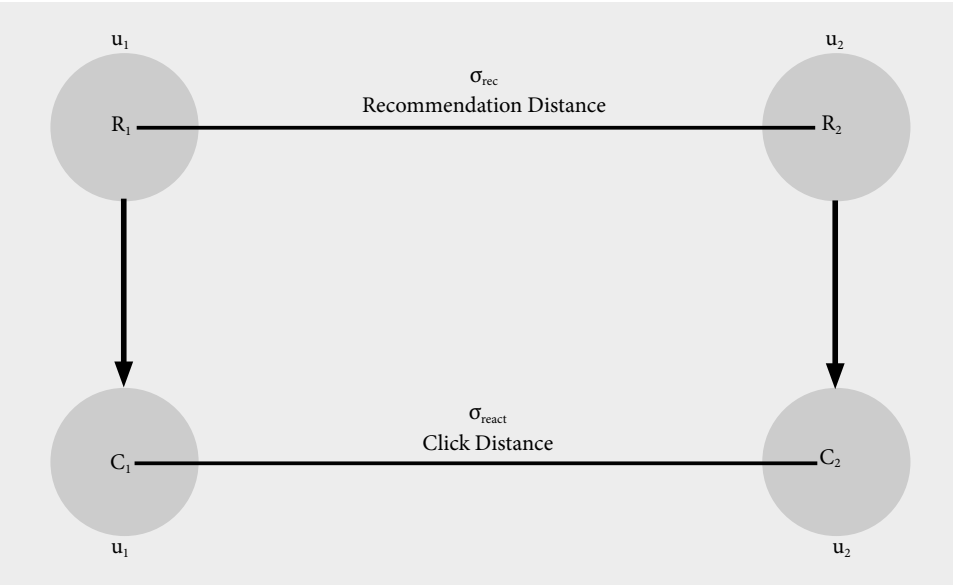


Figure 10.4: The imposed and resultant distances of a personalized recommendation using the recommendations and clicks of two users. Arrows show that recommendations influence clicks. The difference between *RDistance* and *CDistance* must be *0* in a balanced personalized differentiation.

$$d(u_i, u_j) = d(f(u_i), f(u_j)) \tag{10.4}$$

10.3.1. The Pull-Push Score

Figure 10.4 shows the relationship between recommendations and clicks for two users, u_1 and u_2. User u_1 is served R_1 and has consumed C_1. Similarly, user u_2 is served R_2 and has consumed C_2. The arrows from Rs to Cs show the coupling—the direction of influence of recommendations on clicks.

If recommendation lists for two users differ, that difference is the result of (personalized) differentiation by the recommender system. Users will click on some and not on others, and thus will have different click vectors from their respective recommendation vectors. The difference between the clicks is the result of the difference imposed by the RS and the difference created by the users' own selection. The discrepancy between the difference in recommendations and difference in clicks shows the magnitude of push or pull. We call the discrepancy score between the differences in recommendations and the difference in the user reactions a pull-push score.

The pull-push metric computes 1) the distance between the recommendation vectors (σ_{rec}) to quantify the difference imposed, 2) the distance between the click vectors (σ_{react}) to quantify the resulting difference and 3) the difference between the distances to obtain the pull-push score. Mathematically, for a pair of users u_i and u_j, we define σ_{rec} in Equation 10.5 and σ_{react} in Equation 10.6.

$$\sigma_{rec_{u_i u_j}} = d(R_{u_i}, R_{u_j}) \tag{10.5}$$

$$\sigma_{react_{u_i u_j}} = d(C_{u_i}, C_{u_j}) \tag{10.6}$$

The σ_{rec} is the distance that the system estimates and maintains between the two users. Not all recommendations result in clicks. Within the imposed differentiation, users have the freedom to consume some items and ignore others. For example, a certain RS may recommend to u_1 and u_2 some number of (shared) items on Joe Biden and on Donald Trump and some members of unshared items. The two users may ignore the shared items of Joe Biden and Donald Trump and read only the respective unshared items, or they may only read the shared items and ignore the unshared items. In the first case, the users would be pushing each other away, signaling that the items in the overlap are not of interest to them. In the second case, they would be coming together, "telling" the system that they like shared content more than the other items.

σ_{react} is the measure of how different the users are in terms of the content they choose to consume given the distance imposed by the recommender system. Using both σs, we define δ (the pull-push score), in Equation 10.7, as the difference between σ_{rec} and σ_{react}.

$$\delta_{u_i u_j} = \sigma_{rec_{u_i u_j}} - \sigma_{react_{u_i u_j}} \tag{10.7}$$

10.3.2. Properties of the Pull-Push Metric

When σ_{rec} is 1, the pull-push metric is undefined. By definition, a σ_{rec} score of 1 means that the pair of users are in a state of absolute isolation from each other. In this state, they have no shared items and we can not infer a score that is meaningful. For any Recommendation Distance (RD) score in the interval $[0, 1)$, the Click Distance (CD) score falls in the interval $[0, 1]$.

For any $\sigma_{rec} \neq 1$, the δ score falls in the interval $[\sigma_{rec} - 1, \sigma_{rec}]$. Let us consider two special cases, no personalization and extreme personalization, to illustrate this. For no (personalized) content differentiation, $\sigma_{rec} = 0$. The δ range for this is $[0 - 1, 0] = [-1, 0]$. For an extreme content differentiation σ_{rec} score is closer to 1, which means the RS has served nearly completely different content to the pairs of users. The δ range for this is $(1 - 1, 1] = (0, 1]$. The σ_{rec} score determines the range of δ. This is due to the fact that the user reactions are contingent upon the recommendations. If a RS has imposed a distance of 0.8 between a pair of users, the users have a maximum further distance of 0.2 to be more different. They can not be more different than that since they can not go beyond mutually exclusive consumption. They can, however, be as similar as they want by the amount of shared content they choose to consume. Conversely, if the RS does little or no content differentiation between two other users, the users have a large possibility of consuming as different content as they want, but a smaller possibility of being more similar.

The potential δ scores for all pairs of users in a RS fall in the interval $[-1, 1)$. Let us consider two special cases of σ_{rec} and σ_{react} to determine the bounds of the interval. For a pair of users with no content differentiation between them, $\sigma_{rec} = 0$. Given this, let us assume the user reactions result in a $\sigma_{react} = 1$, which means users ended up consuming exclusively different content. The δ score is $0 - 1 = -1$. This means the RS predicted the

users to have exactly similar interests; the users, however, showed that they have as different interests as it can be. Now consider the second case, where the σ_{rec} score is almost 1, which means the RS has served nearly completely different content to the pairs of users. Given that, the maximum possible δ score is close to 1. This is when the RS predicts the users are nearly mutually exclusive; the users, however, show that they actually have exactly similar interests. When we combine the highest possible and the lowest possible δ scores, we obtain $[-1, 1)$, which is the interval for the potential δ scores for all pairs of users. The practical score depends on the maximum and minimum content differentiation that a RS imposes between pairs of users.

10.3.3. Interpreting Pull-Push Scores

A δ score shows the magnitude of difference that must be either avoided or imposed, depending on whether it is negative or positive, to arrive at the balanced differentiation. A δ score of 0.5 shows the need to avoid a distance score of 0.5 between the pair of users in question. A δ score of -0.5, on the other hand, shows the need to impose an additional distance score of 0.5 between the users. A balanced differentiation is one that results in $\delta = 0$. While σ_{rec} is the current level of differentiation, σ_{react} is the differentiation that the users want given the σ_{rec}, and δ is the amount of differentiation distance that needs to be added to or decreased from the σ_{rec} in order to achieve σ_{react}.

The pull-push score quantifies the tendency of the users' responses given the RS's differentiated recommendations. When δ is positive (Pull), the tendency of the pair of users is pulling each other or coming together. It indicates that users find the distance imposed in the recommendations more than necessary. A positive score then is the degree of disapproval, by the users, of the RS's imposition of unnecessarily larger difference between them. We see this as a measure of protest, by the users, at the imposed difference by choosing to consume a larger proportion of the shared content.

If δ is negative (Push), the tendency of the pair of users is drifting apart. Drifting apart signals inefficient content differentiation; their drifting apart is their attempt to avoid (protest) it. A Push is a measure of disapproval, by the users, of the lack of enough content differentiation by the RS.

The δ score is defined for a pair of users. We can aggregate δ scores to obtain group- or system-level averages. But we need to be careful not to sum up positive and negative scores. By averaging all positive δ scores, we obtain the average degree of disapproval, by the users, of the RS's unnecessary imposition of difference, and by averaging negative scores we obtain the average degree of disapproval of the lack of content differentiation. Note that a recommender system can, at the same time, be disapproved for not doing enough content differentiation for some users and for doing over-personalization for some other users. It can even be disapproved, by a single user, for over-personalization with respect to some other users, and under-personalizing with respect to some other users. A user having a negative δ score with one user, and a positive δ score with another user shows they are experiencing both over- and under-personalization. When all pairwise δ scores are 0, the RS is said to have achieved a balanced level of content differentiation for all users.

If a certain user-pair has a δ score different from the balanced state of $\delta = 0$, it shows that there is a potential for improving the recommendation system, by either increasing or decreasing the level of content differentiation.

10.4. Application on News Recommendation Datasets

In this section, we demonstrate applications of the metric on news recommendation. Large datasets from news aggregators that provide personalized news recommendation and the resulting user reactions would be especially suited for experimentation. But such datasets are hard to come by. Instead, we chose two news recommendation datasets: simulated and real-world. We use the simulated dataset to explore and show some interesting aspects of the pull-push metric. The real-world datasets has some limitations that we will explain later. But, first we discuss some practical considerations and choices.

10.4.1. Selection of Vector Components and Users

The components for the recommendation and click vectors can be either items served, meta data about the items, named entities or other relevant features. For our experiment, we use items for components in both the simulated and the real-world cases. The values of the vector components are, however, different. For the simulated case, the values are Boolean *0*s and *1*s. In the real-world case, they are counts of the number of times an item is shown and clicked.

The users can also be of different granularity, such as the individual user, or cohorts of users such as a demographic group, or a geographic unit. In the simulated case, we used individual users. In the real-world case, we used a higher level granularity of geographic units as users. The choice of the geographic unit for the real-world datasets has two advantages. The first is that we wanted to overcome the data sparsity problem since our dataset is not big enough. News recommendations result in less than 1% of the recommendation being clicked, which together with a small dataset can be a big sparsity problem giving the impression that many users have the same reaction lists. The second is that it offers us the opportunity to quantify the RS behavior at higher user-granularity than the oft-used level of individual users. There is evidence that geography [99], demography [133] and educational background [134] affect the consumption of information. Given these limitations and opportunities, we found applying the pull-push metric at cohort-level both overcomes certain problems of our available dataset and offers opportunity.

10.4.2. Selection of Distance Metrics

One can use different distance metrics depending on one's tastes and goals. It is important to normalize the vectors such that σ_{rec} and σ_{react} are comparable. In this chapter, we experimented with two different distance metrics. For the simulated case, we used Jaccard distance, which measures the difference (dissimilarity) between sets. When aggregating users into groups, one might be interested in viewing recommendations and reactions as distributions instead of as sets. We do exactly that in the real-world application, where we use the Jensen-Shannon Divergence (JSD) distance metric, which is suited for comparing distributions.

10.4.3. Application on Simulated Datasets

First, consider the user recommendation list and the reaction list as sets. For a distance metric we use the Jaccard Distance Metric, given in Equation 10.2. Table 10.1 shows a simulated data of users, sets of recommendations and the resulting sets of user reactions. We use

Table 10.1: Simulated data of a RS's recommendations showing users, sample sets of recommendations, and sets of the resulting user reactions. R and C stand for recommendations vector and click vector respectively, and u stands for user. The value *0* represents shown (clicked) or and the values *1* represents not shown (not clicked).

Items	u1		u2		u3		u4		u5		u6		u7	
	R	C	R	C	R	C	R	C	R	C	R	C	R	C
item1	1	1	1	0	1	0	0	0	1	1	1	1	0	0
item2	1	1	1	0	1	0	0	0	1	1	1	1	0	0
item3	1	1	1	0	1	0	0	0	1	1	1	0	0	0
item4	1	1	1	0	1	0	0	0	1	1	1	0	0	0
item5	1	1	1	0	1	1	1	1	1	1	1	0	1	0
item6	1	0	1	1	1	1	1	1	1	0	1	0	1	0
item7	1	0	1	1	0	0	1	0	1	0	0	0	1	0
item8	1	0	1	1	0	0	1	0	1	0	0	0	1	0
item9	1	0	1	1	0	0	1	0	1	0	0	0	1	1
item10	1	0	1	1	0	0	1	0	1	0	0	0	1	1

this simulated data to highlight some possible pull-push scores that can show interesting relationships between recommendations and the resulting user reactions.

Let us consider a few pairs of users that are interesting. Users u_1 and u_2 are recommended exactly the same set of items. The Jaccard distance between their recommendation lists is, therefore, *0*. Given these exact sets of recommendations, the users ended up consuming exclusively different content. The Jaccard distance between their clicks is, therefore, $\sigma_{react} = 1$. The pull-push score for this pair of users is $0 - 1 = -1$. The RS predicted that this pair of users has exactly the same interests; the users disagreed by consuming as different content as possible.

By contrast, users u_3 and u_4 (see Table 10.1) are recommended nearly mutually exclusive sets of items, that is, a distance score of $\sigma_{rec} = 0.8$. Despite that, they ended up consuming exactly the same sets of items, resulting in $\sigma_{react} = 0$. This gives a pull-push score of *0.8*. This positive pull-push score is the magnitude of unwanted difference imposed between them. According to the users themselves, less differentiation of content between them would have been better to meet their interests. In this case, the RS predicted a distance of *0.8*, but the users wanted a distance of *0*, a state of no content differentiation between them.

The cases of the two pairs of users, u_1 and u_2, and u_3 and u_4, show users turning out completely the opposite of what the RS predicted them to be. Users can also protest the RS's differentiation by ending up more different or more similar than the RS makes them to be. For instance, users u_6 and u_7 are recommended the same sets of items as users u_3 and u_4. In both cases $\sigma_{rec} = 0.8$. The user reactions are, however, very different. In the case of users u_3 and u_4, they ended up consuming exactly the same sets of items, despite the large content differentiation imposed by the RS. In the case of users u_6 and u_7, they ended up consuming even more different items, resulting in a pull-push score of $\delta = -0.2$. Given exactly the same sets of highly differentiated recommendations, one pair of users ended up consuming exactly the same content, and another pair asked for even more content

differentiation, a mutual exclusivity. Similarly, given highly similar recommendations, users can end up consuming either mutually exclusively different or even more similar items.

Users u_1 and u_5 are recommended exactly the same sets of items, and they have also consumed exactly the same sets of items. The pull-push score for this pair of users is 0.0.

The simulated examples above show some possible differentiation and some possible resulting user reactions. Users agree or disagree with the RS's differentiation by the magnitude and sign of the pull-push score.

10.4.4. Application on Real-World Datasets

For this part of our experiment, we use datasets of recommendations and clicks collected from a real-world recommender system's recommendations and reactions. We view the recommendations and the clicks as distributions, as opposed to sets because 1) the counts of recommendations and clicks are not suited for set-based processing 2) sometimes there is a need to treat recommendations and clicks as distributions and 3) we want to use this opportunity to show a distribution-based distance metric.

Our datasets were extracted from user interaction history in the Plista platform, which is a recommendation service provider that offered the Open Recommendation Platform (ORP)[1]. The platform brought together online content publishers in need of recommendation services and news recommendation service providers that provide recommendations by plugging their recommendation algorithms to the platform. The recommender we used is a simple recency-based recommender that does little content differentiation as it recommends the most recent/popular items.

Several media outlets were present in the consumption of recommendation services. For our analysis we choose two popular German news and opinion portals: Tagesspiegel[2] and KStA[3]. For user groups, we chose the 16 states[4] of Germany.

Data Pre-processing

For each state of Germany, we prepared recommendation and click vectors. The components of the vectors are items and the values are the number of times the items have been shown and clicked in the state. A sample of the recommendation and click vectors for two states, Berlin and Bavaria, are shown in Table 10.2. We prepared such vectors of recommendation and clicks for each of the 16 states of Germany, which results in 16 pairs of recommendation-click vectors, one pair for each German state. The union of all the items that appeared in the recommendations in any of the geographical regions were used as the vector components, thus harmonizing the vector components across all states. Then, we used add-one smoothing in both vectors. The vectors are then converted to conditional probabilities of $recommendation \mid state$ and $click \mid state$ by dividing the vectors by the sum of all recommendations and the sum of all clicks, respectively. As two conditional probabilities that have the same dimensions, they are normalized for the computations and comparisons. Using these conditional probabilities, we compute the pull-push score for each pair of states.

[1] https://www.plista.com/wp-content/uploads/2017/07/plista_ORP_final.pdf, accessed in August 2017
[2] www.tagesspiegel.de
[3] http://www.ksta.de
[4] https://en.wikipedia.org/wiki/States_of_Germany, in July 2023

Table 10.2: A sample of recommendation and click vectors for Berlin and Bavaria before smoothing and normalization.

Item ID	Berlin		Bavaria	
	Reco	Click	Recom	Click
139831852	2800	3	5000	112
138367374	4000	1	376	20
139820561	178	24	804	18
140465405	657	249	1060	10

Application

For a distance metric, we use Jensen-Shannon Divergence (JSD), defined in Equation 10.8. JSD is a symmetric and normalized distance metric based on KL-divergence, defined in Equation 10.9. JSD is suited for calculating distances between distributions. After estimating probability distributions for recommendations and clicks, we use JSD to compare how different they are.

$$JSD(X,Y) = \sqrt{\frac{1}{2}KL(X, \frac{(X+Y)}{2}) + \frac{1}{2}KL(Y, \frac{(X+Y)}{2})} \tag{10.8}$$

$$KL(X,Y) = \sum_i x_i \ln \frac{x_i}{y_i} \tag{10.9}$$

Using JSD as a distance metric, we applied the pull-push metric to the recommendation and click vectors of the 16 German states. We did this for the two news portals of Tagesspiegel and KStA. The pull-push scores for the two publishers are all negative, averaging -0.224 for Tagesspiegel and -0.213 for KStA. Both scores indicate the lack of enough content differentiation, which is unsurprising for a recency-based recommender system. The smaller average score for KStA can be explained by the fact that KStA has a more geographically local readership as compared to Tagesspiegel which is a nationally read portal [99]. A more local readership means less geographical diversity in interests and preferences.

The pairwise pull-push scores for a sample of 10 states[5] are presented in Table 10.3. The upper diagonal shows the pull-push scores for Tagesspiegel and the lower diagonal for KStA. The diagonal cells which are left empty are 0 because there is no pull-push between a state and itself. When we compare the pull-push magnitudes for the same pairs of states in Tagesspiegel and in KStA, we find that, in most cases, their scores in Tagesspiegel are greater than those in KStA. For instance, the pull-push score for Berlin-Saarland is -0.404 in Tagesspiegel, while it is -0.194 in KStA. The magnitudes of the scores indicate that the need for more content differentiation between the state of Saarland and the state of Berlin in the news portal of Tagesspiegel is greater than in the news portal of KStA. The two largest pull-push scores are, however, found in KStA, and they are Westphalia-Bremen (-0.569) followed by Westphalia-Mecklenburg (-0.561). These pairs of states have shown the largest interest for more content differentiation.

The pairwise pull-push scores are presented in the Multidimensional Scaling[6] visualiza-

[5]We show a sample of 10 because the 16 states do not fit on a page.

[6]https://scikit-learn.org/dev/modules/generated/sklearn.manifold.MDS.html, accessed in August 2022

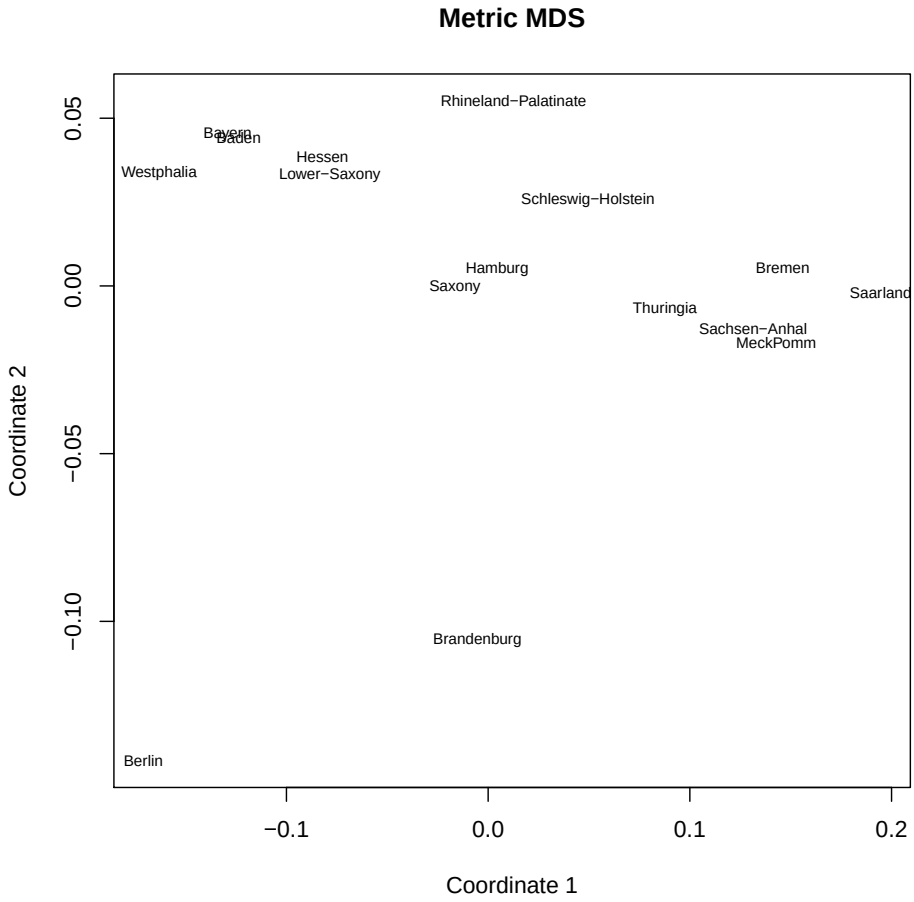
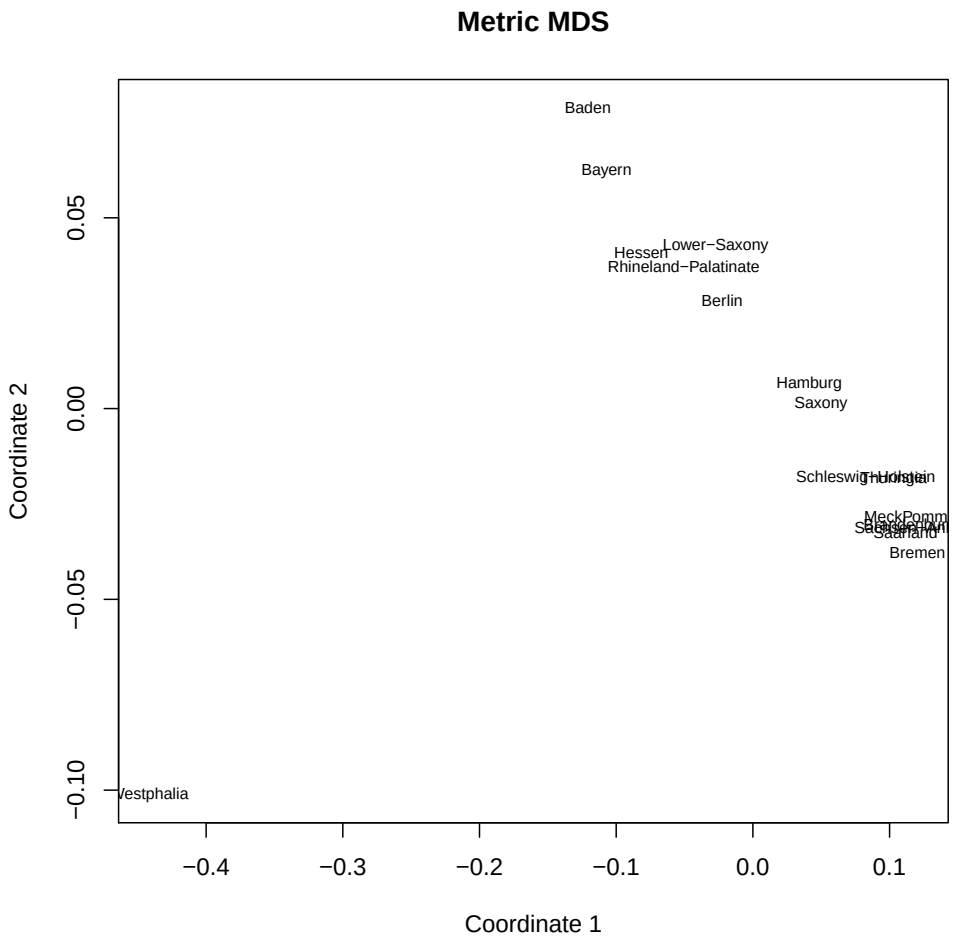


Figure 10.5: A multidimensional scaling of the pull-push scores for Tagesspiegel. The visual distance between states is proportional to the magnitude of the pull-push score. We observe that the highest distances are Berlin-pairings followed by Brandenburg-pairings.



10

Figure 10.6: A multidimensional scaling of the pull-push scores for KStA. Visual distances between states are proportional to the magnitude of pull-push scores. We observe that the highest distances are between Westphalia and the other states.

Table 10.3: Adjacency matrix of pull-push scores for select *10* states of Germany. Bav, Ber, Bre, Hes, Sax, Ham, Mec, Saa, Thu and Wes stand, in that order, for Bavaria, Berlin, Bremen, Hessen, Saxony, Hamburg, Mecklenburg-Vorpommern, Saarland, Thuringia and Westphalia. The part above the diagonal is for Tagesspiegel and the part below the diagonal is for KStA. Comparing the corresponding scores for Tagesspiegel and KStA, we observe that, for most pairs, the absolute value of the scores in Tagesspiegel are larger than those in KStA, an indication that overall there is more potential for personalization in the former. Individually, Westphalia in KStA has the largest absolute pull-push scores, an indication that there is a bigger potential for personalization in this state than in any other states.

	Bav	Ber	Bre	Ham	Hes	Mec	Saa	Sax	Thu	Wes
Bav		-0.221	-0.307	-0.201	-0.146	-0.306	-0.349	-0.196	-0.263	-0.129
Ber	-0.187		-0.366	-0.264	-0.233	-0.359	-0.404	-0.256	-0.321	-0.204
Bre	-0.271	-0.204		-0.206	-0.269	-0.161	-0.156	-0.231	-0.174	-0.334
Ham	-0.215	-0.158	-0.149		-0.176	-0.212	-0.245	-0.166	-0.185	-0.221
Hes	-0.167	-0.164	-0.247	-0.184		-0.271	-0.31	-0.177	-0.227	-0.156
Mec	-0.267	-0.192	-0.087	-0.134	-0.235		-0.165	-0.228	-0.173	-0.334
Saa	-0.267	-0.194	-0.087	-0.137	-0.235	-0.086		-0.265	-0.194	-0.377
Sax	-0.218	-0.158	-0.138	-0.126	-0.188	-0.125	-0.125		-0.195	-0.21
Thu	-0.257	-0.185	-0.098	-0.127	-0.227	-0.083	-0.091	-0.121		-0.288
Wes	-0.389	-0.452	-0.569	-0.502	-0.401	-0.561	-0.56	-0.509	-0.554	

tions in Figure 10.5 and Figure 10.6 for easy viewing. The scalings are a way of visualizing data points in a high-dimensional space by mapping them to a lower-dimensional space, by preserving the relative (not actual) distances between them. As can be observed from the figures, the states in Tagesspiegel are generally more scattered, indicating a larger gap between the states' needs for content differentiation and current differentiation levels. Looking at specific pairwise scores for Tagesspiegel, we find that Berlin-pairings show greater push. Next to Berlin, Brandenburg-pairings show greater push. A possible explanation is the fact that Tagesspiegel is primarily a local portal for Berlin and Brandenburg, whose pull-push distance is not very big, as can be seen from the figure. As such, Berlin and Brandenburg are showing a need for a more differentiated content, probably more local recommendations, as opposed to the other states which would probably be more interested in the more national-level news.

For KStA, Westphalia-pairings exhibit the tendency to differentiate from each other. This can also be explained by the fact that KStA is based in Cologne and most of its readers come from Westphalia (the state with capital Cologne) [99]. That means that the scores indicate more need for content differentiation between Westphalia and the other states. As Tagesspiegel is local to Berlin and Brandenburg, so is KStA for Westphalia.

The content differentiation needs, indicated by the pairwise scores in Tagesspiegel and in KStA, are consistent with intuitions and previous findings of the impact of geography on news consumption [99]. For example, both Tagesspiegel and KStA have local news categories and a previous report [99] showed that those categories are mainly read by Berlin (and Brandenburg) and Cologne (Westphalia), respectively. Apparently, the recommender system has not captured these trends, and hence the bigger push scores we observe.

The pull-push scores give the opportunity to observe the impact of content differentiation at both aggregate and pairwise levels at the chosen user granularity. At the aggregate level, it shows the degree of differentiation success at the level of the system or the section of users. At the pairwise level, it shows a fine-grained score at the level of pairs of users, opening a possibility for selective intervention, without affecting other pairs. For example, one can

decide to only fine-tune differentiation between Westphalia and the rest of the other states, without affecting the rest of the other pairs.

10.5. Discussion

The pull-push metric is a generic user-centric metric that measures a RS's content differentiation success/failure at meeting user content differentiation needs as a function of the differences between user-pair differences in recommendations and in the resulting user reactions. It is a versatile metric allowing the choices of the vector components or the members of the set, the distance metric and the user granularity—one can use demographic levels, geographic cohorts or gender groups. The ability to produce average scores and pair-wise scores provides RS practitioners and researchers with a useful and practical measurement to zoom in and zoom out to gain insights and to subsequently intervene.

The pull-push measure is related to several other measures. The Pull-Push score concerns itself with content differentiation into recommendation lists. Personalization, however, covers the ranking of the recommendation lists. Below, we show how pull-push's measures on the differentiation of recommendation lists and "the potential for personalization" can be combined to produce what we call "potential for improvement". We also discuss the pull-push score in relation to popularity bias and how the latter can be punished, if need be. In measuring personalization, one comes face-to-face with normative standards on news journalism. We therefore discuss the pull-push score in relation to this too. Finally, we discuss the limitations and weaknesses of the metric.

10.5.1. Potential for improvement

In Section 10.2, we have explained earlier the difference of the pull-push metric with Teevan et al.'s "potential for personalization" [126]. The two metrics concern themselves with different stages of the recommendation process: the pull-push metric with the content differentiation to recommendations lists, and the potential for personalization with the reranking of the recommendation list. When using the potential for personalization, if the user clicks the recommended items in the order of their ranking, then the potential for personalization (henceforth γ) is 0. There are other differences related to the main difference. The potential for personalization does not penalize a RS for showing more irrelevant items as long as the ranking of the clicked items is correct. Recommending 20 items whose three top-ranked items are clicked and recommending 3 items which are all clicked have the same potential for personalization. In the pull-push case, a RS is penalized for showing items that are not clicked.

In our case, there are two cases for improvement. One is when the pull-push score is in push state (negative pull-push score), indicating that the users find the content differentiation between them falling short of meeting their preferences. This can also be viewed as the amount of potential for personalization at the content differentiation stage, as opposed to the ranking stage. There is a potential for personalization for a user in the pull-push case wherever the pull-push score with another user is negative. If we sum up all the negative pull-push scores a user has with other users and average them, then we can say we have the average potential for further differentiated recommendations for that user. If we do the same for all users, we can then compute the average potential for personalization for all users.

The pull-push and Teevan et al.'s potentials for personalization differ, but they compliment each other. Both indicate potentials for differentiation (personalization), but at two different stages (the selection of a recommendation list and the ranking of the recommendation list) of the recommender system. They can be combined (assuming equal weight) as in Equation 10.10 to obtain the total potential for personalization (PP) at both stages. We use $|push|$ because push is negative, and the sign is not needed in this case. By dividing the score by two, it would fall in the interval $[0, 1]$ where 0 indicates no potential for personalization at both stages and 1 indicates the highest possible potential for content differentiation. The highest potential is when $\delta = 1$ and $\gamma = 1$. Essentially, therefore, the amount of potential for personalization in this combined sense is the amount of further effort/struggling needed to select items the users want and to rank them according to how the user would click them.

$$pp = \frac{|\delta_{neg}| + \gamma}{2} \tag{10.10}$$

The other potential for improvement in the pull-push case is when the pull-push score is pull state (positive pull-push score), indicating that users find the content differentiation more than that is necessary. Since this score is not the desired balanced state, it can be seen as a potential for improvement. We refrain from calling it a potential for personalization because a positive score indicates the need for less personalization. In the sense of effort needed to bring the recommendation list to what the users want, however, it represents a potential for improvement. Since positive score, negative score or the potential for reranking all indicate the need for further efforts to satisfy the user interests, we can combine them all to obtain the total potential for improvement (pi) as in Equation 10.11. The potential for improvement would again fall in the interval $[0, 1]$. When one needs to take action, however, one needs to first check the sign of the δ score to see whether to do less or more personalization at the level of the production of the recommendation lists.

$$pi = \frac{|\delta| + \gamma}{2} \tag{10.11}$$

10.5.2. The Pull-Push Metric and Popularity Bias

The pull-push metric is in part a precision-oriented metric [135] in that the score calculation starts with the recommendation list (top-N) and the resulting user reaction list. A known problem of top-N-based evaluation metrics is popularity bias, which is the act of rewarding a recommender system that recommends popular items, as opposed to rewarding algorithms that personalize content according to user needs [136, 137].

The dominant opinion is that popularity bias is an undesirable bias and should therefore be removed [136, 138–140]. Cañamares and Castells [137], however, ponder about whether we actually want to get rid of the popularity bias. They ask thus: if recommending popular items happens to be the right thing to do, then should not recommending them be favored and rewarded? Regardless of the opposing views in whether popularity bias should be removed or not, popularity bias is a prevalent property in recommender systems and their evaluation metrics. We would, therefore, like to discuss the pull-push metric in relation to it.

The pull-push metric is susceptible to popularity bias in that a recommender system that recommends popular items can achieve as good a pull-push score as another one that

does good personalized recommendations. For instance, if a certain recommender system recommends the top 3 popular items to two users, and both the users click on three of them, then pull-push score would be 0, indicating a balanced content differentiation. While this may not be a problem for a recommendation provider interested in maximizing clicks, or the user who is content with being recommended popular items, there are situations where one would like to diversify content recommendations or to penalize popularity bias.

One way to penalize popularity bias in the pull-push metric is to look at the σ_{rec} score in addition to the δ score. A higher σ_{rec} shows a higher level of content differentiation (personalization). A recommender system with a higher σ_{rec} score and a lower δ score shows that the recommendations have been differentiated (personalized) and that users are content with the differentiation.

10.5.3. Pull-Push Score, Normative Standards, Filter Bubble and Fairness

Developing a metric to quantify recommendation success is tricky because it touches on the doubly contested area of normative standards for journalism. Normative standards for human journalism are pluralistic, and already contested. Normative standards for algorithmic journalism are doubly contested [3]. As journalism is a normative activity, scholars state the importance of going beyond descriptive investigation to consider the normative implications (even if it is a contested) of algorithmic recommendation [3]. Encouraged by this call, we attempt to relate the metric to the discussion on normative standards and to ground it on a particular normative framework.

Natali Helberger [141] outlines three democratic models, namely liberal, participatory and deliberative, that are used in assessing media, and she discusses their implication for news recommender systems. In the liberal democratic model, recommender systems put user interests and preferences central stage. Under this model, it is the prerogative of citizens to choose what information they need, and also it is fine for a news platform to provide information items customized to the needs of the user. In the participatory model, the participatory recommender will need to make sure recommendations are fair and inclusive representation of different ideas and opinions in society, in addition to making the user gain a deeper understanding and make them feel engaged. This democratic model operates out of principles to nudge users to "powerful ideas and opinions". In the deliberative model, the media assumes a public forum function where "the different ideas and opinions in a democratic society can be articulated, encountered, debated and weighed".

The pull-push metric, as a user-centric metric of content differentiation effectiveness falls under the liberal democratic model. This means that user-interest takes center stage along with attendant implications. For example, issues of filter bubble, fairness, inclusiveness and diversity will need to be seen from the perspective of user interest and preference. Personalization is a response to information overload. In fact, we can consider the size of the push score as a measure of information overload yet to be mitigated to arrive at the user-preferred recommendation list. Under the liberal normative standard, mitigating this information overload is a necessity. As we over-personalize, however, we risk isolating the user in a filter bubble [13], which is a societal concern. In the liberal model, as long as the user somehow does not see the filter bubble as a problem, it is fine and does not need to be avoided since the user decides for themselves.

Similarly, the presently hot issues of bias and fairness [142–144] will only be considered from the perspective of user interest. Recommender systems are multi-stakeholder environments. Fairness notions may contradict not only utility, but also fairness notions of the different stakeholders. What is fairness in this context, fairness for whom, and according to whom? Castillo [143] views algorithmic fairness in ranking from the point of view of the people and organizations that are being searched. Edizel, et al. [145] views fairness in recommender systems from the point of view of users. Burke [146] proposes that recommender systems have different fairness requirements for the different stakeholders. News recommendation is different from other recommendations because items are ephemeral, and unlike recommendations whose aim is post-click conversions (buying an item or booking), news recommendation ends with clicking and maybe dwell time. Defining fairness in terms of the predictability of sensitive attributes, as did Edizel, et al. [145], does not account for discrimination on legitimate grounds such as on the basis of different base rates [147]. For example, Edizel [145] presents two specific reddit threads, "makeupaddictions" and "cscareerquestions", whose 97% and 84% of their comments are submitted by females and males respectively. It does not make sense, in the liberal democratic model, to eliminate the predictability of gender from the recommendation matrix involving these threads. If the pull-push score is 0, meaning a balanced personalized recommendation for each user according to the user themselves, can the recommender system be considered unfair? We surmise that while a non-balanced pull-push score may not necessarily indicate a measure of unfairness, a balanced score does not imply unfairness, as long as the user perspective is concerned.

10.5.4. Limitations of the Pull-Push

The pull-push measure assumes enough presence of shared items between the recommendations lists for the users to be able to diverge or converge. When over-personalization is so high that the generated recommendation get closer to disjoint sets, the pull-push scores are highly distorted. Interpreting a pull-push score in this situation can be a bit problematic. We consider this as one limitation.

Another limitation is data sparsity. Since reactions (for instance clicks on items) are a small fraction of the recommendation list, pull-push is susceptible to data sparsity. In our case, we have minimized this by using the higher user-granularity of a geographic region, but data sparsity at the individual user level may be a bigger problem. We recommend the use of large datasets to minimize the impact of data sparsity.

Although not infeasible to integrate in future work, at the moment, the metric does not consider time. We have used a batch dataset, but maybe a time series analysis could be more appropriate. Finally, we do not consider our dataset the best possible choice for analysis using this metric, because recommendation was recency based, meaning doing almost no personalization, leading to negative scores. That is why our scores are all negative indicating push. An additional dataset of recommendations lists and the resulting user reactions from a recommender system that actually implements some personalization would be very good to compare it to our dataset, but such datasets are hard to come by in the public domain.

10.6. Conclusion

Personalized recommendation list generation can match users with items of their interest and lead to increased engagement, but it can also mean that users may end up more differentiated beyond what they want. To see whether our differentiated recommendations fall short of the ideal content differentiation or goes beyond what is necessary, we introduced the pull-push metric to measure the success of a RS at generating user-preferred recommendation lists. The metric quantifies the degree of pull or push between users using the difference in recommendation lists and the difference in the resulting reaction lists.

In the pull-push measure, the production of personalized recommendations is viewed as the act of imposing some difference (distance) between pairs of users in terms of the items they are recommended to. This view of content differentiation is carried to the pull-push metric offering a novel way of measuring a RS's content differentiation effectiveness in meeting users' differences in interests and preferences. The metric is an abstraction from the actual recommendation lists and the resulting users' reactions; it concerns itself with the preservation of the differences or similarities introduced at the recommendation stage in the resulting user reactions. The pull-push metric is suited to the practical exploitation of a recommender system, as it can be used to get insight at different granularities and to suggest a course of action at any stage of the deployment of a recommender system. The metric is versatile allowing choices in distance metric, user granularity, vector components and their values. With appropriate normalization of the vectors, pull-push distance scores fall in an interval with endpoints, making it possible to compare different recommender systems' personalized differentiation success.

We applied the method to simulated and real-world datasets, using different distance metrics and different granularity of users. In the simulated case, we used the set-theoretic distance metric of Jaccard distance and individual users and in the real-world one, we used Jensen Shannon distance (a distance metric for probability distributions) and cohorts of users of geographical units. The abstraction to geographical cohorts of users provided us with two advantages: less data sparsity and an opportunity to examine (personalized) content differentiation at a higher user granularity. In the simulated case, we showed different interesting scores of the pull-push score and their interpretations. In the real-world experiment, the pull-push score suggested the presence of under-personalization (explainable by the importance of recency in news RS), and therefore a potential for more personalization when the desire is to satisfy user interests.

We have discussed the metric in relation to other very related metrics on personalization and shown how it can be combined with the "potential for personalization" to serve as a metric of the potential for improvement in the content differentiation and the ranking stages of a RS. Also, we discussed how the metric is susceptible to the known popularity bias of recommender systems and offered a way to penalize the bias in case one wants to do so. We also discussed the metric in relation to normative standards, and fairness in recommender systems. Finally, we discussed the limitations of the pull-push metric.

The pull-push metric is user-centric and in tune with liberal normative vision and the fundamental tenet of personalization, that is, that users have differences in information interests and preferences. In the pull-push measurement, a balanced (personalized) content differentiation is one where the proportion of imposed differences during recommendation are approved in the resulting user reactions. If that is not the case, it indicates a degree of

disapproval of the RS's under-personalization (push) or over-personalization (pull) by the users themselves.

11

Conclusion

This dissertation covers research into the different stages of the recommender system pipeline. We started with the exploration of approaches to a special type of recommendation task, that of CCR, and delved into the investigation of the stages of a recommender system pipeline, to further the understanding of the components of the recommendation pipeline and the impacts of choices on the overall system.

In particular, we investigated approaches to the task of CCR, where we explored string-matching methods where entities are represented with an expanded set of labels and machine-learning approaches that try to combine our best entity representation and best-performing algorithms. We then proceeded to investigate the interplay between feature sets and machine learning algorithms. We demonstrated the inherent interplay between the feature sets used, the algorithms selected and their impact on the conclusions we make. Following this, we focused on the filtering stage of the recommendation pipeline and its implications on the later stages of the pipeline and overall performance. We maximized for recall, establishing an "upper-bound" recall for the filtering task in CCR and conducted an error analysis to identify the causes of some documents not being retrieved.

Next we focused on News Recommendation. Our work on news recommendation involved a multi-perspective investigation of news recommenders and their evaluation. Unlike in CCR, where the target user is an entity represented by a KB profile such as a Wikipedia article and the items to be recommended are considered for their citation-worthiness, the user in news recommendation is an actual individual that reads news and the items are recommended to satisfy the user's interests and preference. This poses different challenges and requires a different approach.

We started with an investigation of the influence of geographical information on news consumption and a comparison of recommender systems in an online setting, where we also incorporated geographical information for improving recommendation lists. This research was carried out in a real-world setting where news providers and recommendation providers were brought together in a common platform. Upon observing discrepancies in our initial experiments and in reported performance differences of new recommendation algorithms, we conducted a multidimensional investigation of news recommendation evaluation to un-

derstand the factors affecting performances and evaluations. We quantified and classified causes of "random" or idiosyncratic performance differences in online recommender system evaluations, evaluated recommender systems in offline and online settings, and investigated recommender system performance (in)consistencies across years. We highlighted the inconsistencies and problems in reported news recommender performances in different evaluation settings and advised against accepting any reported improvements as real improvements as there is a significant performance difference as a result of idiosyncratic and "random" causes not related to the algorithm's performance.

Finally, we zoomed in on the broader role of recommender systems in society. Recommender systems are seen, on one hand, as mitigating information overload, and on the other hand isolating users in filter bubbles, which are only partially their own choices. This debate, measured from a societal perspective, assumes the presence of overpersonalization. However as the degree of personalization of recommender systems is hard to measure, the arguments lack quantification and tend to be opinionated. To help address this, we proposed a user-centric metric for quantifying the degree of over- or underpersonalization of a recommender system from the user perspective. We demonstrated the metric on simulated and real-world datasets. We believe this contributes to the debate on the impact of personalized recommendations in society by providing a method for quantifying the degree of personalization.

11.1. Main Findings

We present the main findings of this thesis by providing answers to the research questions posed in the introduction chapter grouped by research themes.

11.1.1. Theme I: Cumulative Citation Recommendation

Knowledge Base Acceleration aims to improve the knowledge base maintenance and curation process by developing systems that automate the preparatory tasks and leave the final reduced task to be executed by human curators with less effort. Its sub-task, CCR, is defined as, for pre-selected KB entities, filter, rank and recommend items, from a stream of documents, according to the relevance (citation-worthiness) of the documents to the KB entity profiles. Three broad areas have been investigated under this theme.

11.1.2. Approaches to KBA-CCR Task

We first explored two approaches to the KBA-CCR task. Two principal approaches, string-matching approaches [27] and machine learning approaches [148]. Different variants of string-matching and two multi-step machine-learning approaches were employed. We first asked the following research question:

RQ1 How do simple string-matching approaches to the CCR task perform?

Using rich entity representations from a resource called Google Cross-Lingual Dictionary (GCLD), we experimented with a string-matching approach where we aimed to separate the central + relevant documents from the garbage and neutral ones. This approach performed well achieving one of the best results in comparison to other participating systems in the TREC KBA task. We observed that two factors affect the task of CCR: entity

representation and the scoring function used. A very important point is that entity representations should be used as they are i.e. without lower-casing and without stripping off punctuation. The importance of scoring was observed by different scoring functions which resulted in different results.

The GCLD resource we used for entity representation is a mapping, with a probability distribution, from strings to concepts and vice versa. The approach of representing entities with rich labels and alternative names from the GCLD achieved good performance in general, but it was especially good at recall. Having observed that the state-of-the-art approaches to the KBA task employed multi-step classification and learning-to-rank approaches, we decided to experiment with these, among others, by using our rich entity representation for feature engineering. We sought to answer the following research question.

RQ2 Does the use of the rich entity representations from our string-matching approach with machine learning approaches result in improved performances?

Using different feature sets in classification algorithms J48 and Random Forest, we experimented with different variations of training datasets. For entities for which rich representations exist, and for the combined central+relevant documents, the machine-learning approach achieved competitive results. The approach, however, was weak in identifying citation-worthy information for Twitter entities and lesser-known Wikipedia entity groups. This is expected since almost none of the rich features did apply to them.

Generally, our machine-learning approaches to the CCR task in the 2013 iteration of the TREC KBA track did not do well, especially in relation to the fact that the rich entity representations we used did well in the 2012 track. One reason is due to the change of the CCR task in TREC KBA 2013, especially the introduction of less-known Wikipedia and Twitter entities. They achieved competitive results for well-represented entities, but even on that, the string-matching approach to the TREC KBA task was better. The performance and simplicity of the string-matching approach also make it more attractive.

To the best of our knowledge, there are no research works directly focusing on the research question addressed here. There are, however, relevant new developments in neural approaches and pre-trained models. The CCR task can be cast as a text ranking problem where documents are ranked according to their relevance to the preselected entities, and deep learning approaches or pretrained transformers can then be used to accomplish the task. Alternatively, state-of-the-art neural-based entity-mention detection and entity-linking tools can be applied to accomplish some stages of the CCR pipeline. Flair [73] which is used as a state-of-the-art entity-mention detection tool, end-to-end entity linker RefiNED [74] based on transformers, REL (Radboud Entity Linker) [75] which uses Flair for entity mention detection, GENRE [76] and BLINK [77] both using fine-tuned BERT architectures can be employed. A more promising approach might be to use neuro-symbolic approaches (combining neural representations, string-matching approaches and knowledge graphs) [85, 86].

There is a relationship between our string-matching approach using rich entity representations performing well and the success of sub-word tokenization methods (including the compression algorithm Byte-Pair Encoding (BPE) [87]) and neural approaches, indicating that the frequency of mention of (parts of) the query terms in a document is related with the document being relevant to the query. Findings about LLM's ability to memorize factual knowledge about entities also support this. Recent studies have shown that the more

frequent mentions of an entity there are in input data, the better that LLMs will memorize facts about the entity and answer factual questions about it [88, 89]. In other words, LLMs find it hard to memorize facts and therefore to answer questions about entities mentioned less frequently in input data. All of these attest to the time-tested but underrated knowledge that the occurrence of query terms in a text is a strong indicator of a document's relevance to the query [90].

The Interplay Between Features and Machine Learning Algorithms
Spurred by our finding in our machine learning approaches that the relative performance ranking of our machine learning approaches contradicted their reported performance ranking in the literature, we abstracted away from the specific KBA-CCR task to investigate the interplay between the choices of feature sets and machine learning algorithms and their impact on relative performances [42]. We asked the following research question:

RQ3 How does the interplay between the selection of features and the choice of algorithms affect performance?

By focusing on feature selection and subsequent choices of machine learning approaches, we investigated the interplay between choices of features and the performance of machine learning algorithms. By varying the number of features and the choice of feature sets based on the contribution of each feature element, we found that 1) a reduced powerful feature set achieves better performance than a large feature set, and 2) the relative performance rankings of machine learning algorithms can vary significantly with the choice of feature sets. The second finding suggests that when comparing recommendation approaches that involve machine learning approaches in their pipeline, performance does not only depend on the choice of the algorithm but also the feature set used. Specifically, we showed that one algorithm's performance ranking can change when the feature sets used are changed. We should therefore be careful in generalizing conclusions when we compare the performance of machine learning algorithms that use different feature sets, and be very careful in claiming the superiority of one particular machine learning algorithm over the other.

The Impact of Filtering and Unfilterable Documents
Following our investigation of the interplay between feature sets and machine learning algorithms, we decided to investigate the first step, the filtering task, in the KBA-CCR recommendation pipeline [149]. We strove to investigate the extent of the impact of so-called "unfilterable" documents on the overall performance. We asked the following research question:

RQ4 How big is the impact of the initial task of filtering in the KBA-CCR overall performance, and what makes documents unfilterable?

We examined the effect of the filtering task from different sides, cleansing, entity profiles, types of entities (Wikipedia or Twitter), categories of documents (news, social, or others) and the relevance ratings (vital or relevant) on recall and overall performance.

We found that cleansing (standard text preprocessing procedures) can result in reducing recall up to 21% by removing (partial) document content. This finding applies more so in

relevant documents than vital (citation-worthy) ones. While this impact of cleansing documents negatively impacts the recall, it has a positive impact on the overall performance of ranking vital documents for Wikipedia entities and a negative impact on the overall performance of the ranking of relevant documents. This suggests that cleansing can be used or not depending on whether the interest is in vitally relevant documents or just relevant documents. This is explained by the fact that centrally relevant documents that usually mention entities by their canonical and/or alternative names are less affected by cleansing as opposed to relevant documents where cleansing can remove related links and advertisement that may contain a (tangential) mention of the entities.

We also found that the types of profiles we choose matter for the filtering task. Canonical representations work better for Wikipedia entities, and partial name variants perform best for Twitter entities. For maximum performance, different entities, relevance grades and document types (social posts and news) should be approached differently.

Our quest in exhaustively examining the filtering stage from different angles is to establish an "upper bound" on recall for the CCR task in question and to identify the circumstances under which documents become unfilterable. Hereafter, we study the accessed relevant documents that have been missed completely when matching surface forms known to us. For our case, we found an upper bound recall of 90%. The 10% unfilterable documents were found to fall into one or more of the following cases: outgoing link mentions, an event becoming relevant for the venue, a related entity mention becoming relevant to the entity, the mention of the main area of operation of an organization becoming relevant to the organization, the mention of group name becoming relevant to an individual member of the group, an artists work becoming relevant to the artist, the mention of a politician becoming relevant to the political party, and sometimes just one needs to have a world knowledge to understand why a document is relevant to an entity. Sometimes it is simply not clear as to why assessors deemed some documents relevant for some entities. While more advanced entity profiling techniques may resolve some of these cases, there will be some documents that are not amenable for automatic filtering, establishing an upper bound on the filtering stage and subsequent parts of the pipeline.

Recent work on token-free neural models operating directly on raw text [93] seems to support the observation that cleansing impacts recall. Specifically, token-free neural models are more robust to noise and perform better on tasks that are sensitive to spelling. Both indicate that the preservation of information in representation helps in improving performance.

Filtering is the first step in KBA-CCR tasks, where the document collection is large and a technical challenge to handle, and it is an important step because it impacts all the subsequent parts of the KBA-CCR recommendation pipeline. In bench-marking initiatives such as TREC KBA where systems employ different pipelines starting with filtering and where systems are compared based on overall performance, either using a uniform filtering algorithm across all participants or a use of scoring where the impact of the used filtering is accounted for can result in a more useful system comparisons. If this can be combined with similar handling of the interplay between feature sets and performance of machine learning algorithms for pipelines that employ machine learning algorithms, for example by uniformizing the feature sets or accounting for their impact on the overall performance system, comparison of systems in bench-marking initiatives can be fairer, clearer and much more useful.

11.1.3. Theme II: News Recommendation

Motivated by the opportunity to study real-world news recommendations involving real users, real items and real businesses seeking recommendations, we started to participate in the CLEF NewsREEL News Recommendation. News recommendation is different in the sense that items are ephemeral, users are mostly anonymous, and there is a need to meet non-algorithmic requirements such as fast recommendation delivery. These features make traditional recommender system approaches such as collaborative filtering unsuitable and instead call for fast and tailored approaches to news recommendation where, in addition to meeting the above challenges, temporal, geographical, and dynamic natures of items and users are considered. Under this theme, we studied two sub-themes.

Factors Influencing News Consumption and Recommendation

We conducted a descriptive study on the role of geography, followed by a study of algorithms in a real-world setting.

We first investigated the role of geographical information in news consumption [99]. This was inspired by an assumption that people are more interested in what happens around them than in what happens in faraway places. For news portals with a large geographical readership, personalizing geographically might be beneficial. We asked the following research question.

RQ5 What is the impact of geographical proximity on the consumption of news?

Using a large history of users interacting with news items gathered from several online information portals, we investigated the relationship between users' interaction and the geographical foci of news portals. We found that, while special interest portals (sport, mechanic, etc) seem to be less geographically localized, the mainstream news portals, on the other hand, exhibit geographical foci. Analysis of the mainstream news portals, by focusing on local categories and the rest, showed a strong relationship between the geographical focus of the news portal and the readership from that geographical focus. The likelihood that a user who reads news from the local category is from the same geographical focus can be predicted reasonably well, especially when higher cut-off values of the user's visit frequency are considered. We used this finding later to devise a geographical recommender system.

In a follow-up study [150], we investigated performance differences of recommender systems in a real-world setting. Our algorithms are largely similar in that they are mainly recency-based (based on recommending the most recently read items), but variations were introduced to study different aspects. One of the variations incorporates our finding above, that the geographical focus of the portal and the readership from that geographical area are strongly related. We asked the following:

RQ6 What are the patterns of news recommenders' performance in real-world news recommendation, and does the incorporation of geographical information improve performance?

Using four interrelated algorithms deployed in a real-world recommendation setting, we sought an answer to the research question posed above. We found that the incorporation of geographical recommendation did not result in a significant performance improvement over

the recency algorithm, of which it is a modification. An interesting twist was the deployment of two instances of the same algorithm to quantify random performance differences in online news evaluation. This showed that the performance of two identical instances of the same algorithm can show statistically significant performance differences, suggesting that caution is needed to take into account performance differences due to random causes in recommender systems evaluations that involve users and news items in a live setting. Care must, therefore, be taken into account when accepting reported performance differences, as those can be due to non-algorithmic random and idiosyncratic factors in the evaluation setting.

Multidimensional Investigation of News Recommendation Evaluation

Here, we answer several research questions focusing on news recommendation evaluation [151, 152]. In particular, we compare the performance of recommender systems online, quantify random performance differences, identify causes for random performance differences and investigate recommender system performance consistencies in several dimensions (offline, online, and across time).

We sought to explain the causes of random performance differences that we observed above and to quantify them. We asked:

RQ7 What are the causes of random performance differences in real-life news recommendations and how can we quantify the extent of random performance?

Motivated by the significant variation in performance of the two instances of news recommender algorithms discussed above, we investigated the possible causes of the "random" performance differences. We identified, classified and discussed the possible causes of performance differences between real-world news recommender systems. These causes are operational biases in the framework or environment in which the recommender systems operate, differences in the sets of users and items, and their interaction.

But even in the absence of obvious causes of performance differences such as operational biases, and users and items observed in the experiment, performance can vary due to other artifacts in the data collected. Such artifacts include, among others, user mood, representational preferences, and other idiosyncratic factors. The presence of these factors highlights the challenges of evaluating and comparing recommender systems in real-world settings. We presented a way of quantifying the effect of random performance in the evaluation by zooming in on the differences in overlap of the results obtained from two competing algorithms that are tested on two settings simultaneously.

The analysis above raised many questions about how to evaluate recommender systems realistically with a validity of the outcomes. We asked:

RQ8 How do news recommender system performances compare offline, online and across periods?

Evaluating the same algorithms in offline and online settings, we attempted to answer part of this question. We run the same algorithm in two time periods (2015 and 2016) to see how their performance estimates compare across periods. Our findings show the lack of consistent performance patterns. Offline performance measurements were not predictive of the online performance measurements, in both absolute and relative sense, and performance scores of the recommender systems in 2015 were not predictive of those in 2016,

both in relative and absolute sense. These findings once again highlight how precarious and uncontrolled the real-world setting is for the evaluation and comparison of recommender systems.

The studies carried out in this theme have been conducted in the CLEF NewsREEL News Evaluation campaign which used the PLISTA Open Recommendation Platform (ORP) to enable recommendation providers and recommendation consumers to interact and communicate over a standardized protocol. Appraising the campaign and platform is necessary, for this explains the context within which the research work has been done. As part of a bigger research work that aimed at an overall appraisal of the CLEF News Evaluation platform [7], we evaluated the platform from the participant's perspective which was the role we had when we interacted with the platform. We asked the following:

RQ9 What are the participant perspectives on the evaluation of their recommender systems in the CLEF NEWSREEL?

Our appraisal identifies two things: opportunities, and user perspectives on accessibility and fairness. The CLEF NEWSREEL provided a great opportunity for the participating researcher to test their recommender systems in a real-world setting. However, participant researchers also have concerns about the fairness of the platform in the evaluations of news recommender systems.

From the participant's perspective, it is possible to conduct a partial investigation into possible operational biases and have a reasonable estimate of the impact of those causes on the performance of a participating system. The participants did have the means to assure themselves of NewsREEL's fairness in the selection of algorithms using only information available from the participant's perspective. We note, however, that an exhaustive investigation of all possible operational biases is a daunting task. Operational biases can happen at the level of pairing some groups of users to different teams or participant systems, pairing some categories of items to some participants or systems, and/or favoring or disfavoring one system based on response and other network factors. The possibility to explore some of the biases was hampered by the fact that participants did not receive direct information on whether their recommendations were clicked.

11.1.4. Theme III: Pull-Push, A Measure of Over- or Underpersonalization in Recommendation

Recommender systems are deployed for different purposes and have different effects. Usually, we want to quantify the effectiveness of a recommender system in meeting a purpose or in causing an intended or unintended effect. In this theme, we focused on measuring recommender system personalization effectiveness [153]. On one hand, recommender systems are viewed as helping people overcome information overload and, on the other hand, they are accused of isolating users in filter bubbles of largely not their own making. Implicit in this debate is the over- or underpersonalization of a recommender system. We propose a metric for measuring this aspect of a recommender system and demonstrate it on simulated and real-world datasets. We asked the following research question.

RQ10 Can we quantify the degree of over- or underpersonalization by a recommender system from a user-centric point of view?

By viewing recommender system personalization as the imposition of a degree of similarity or difference between pairs of users, we proposed a versatile, user-centric metric that can be used to quantify a recommender system's degree of under- or overpersonalization. The metric produces average scores for a recommender system's overall degree of personalization and fine-grained pair-wise scores for users of one's choice of granularity, for example between pairs of cities, or demographic groups. The metric is practical and can be used to suggest a course of action. The metric equips the recommender system researcher or practitioner with a tool that can quantitatively measure, from a user-centric perspective, the degree of personalization and therefore opens a possibility to tackle its effects according to one's interest.

11.2. Future Work

The research work in the thesis can be extended in many ways to tie the different findings together, to understand the recommendation process as a whole better and the evaluation of recommender systems, and to improve performance. Under each theme, we present some potential areas of extension.

11.2.1. Cumulative Citation Recommendation

Under this theme, we have conducted research on the automation of knowledge base curation, called Cumulative Citation Recommendation. We have also specifically zoomed in on the filtering stage of the recommendation process and the interplay between feature sets and machine learning algorithms. One immediate area to build on and expand on is:

Applying deep learning methods for the CCR:
there has been significant progress and improvement in deep learning and pre-trained models since we did our research on CCR. Applying these new developments to the CCR task as a whole or some of its different stages and comparing them with the string matching and (traditional) machine learning approaches is one potential area of future work. Investigating whether casting the CCR task as a text ranking problem and using neural text retrieval, or which stages can be more effectively replaced with which neural approaches and tools and to combine them in a complementary manner with the approaches we used as seen in recent neuro-symbolic approaches is another potential area of improvement. Specifically applying recent neural-based mention detection tools, entity linkers and disambiguators to the CCR task is interesting.

11.2.2. News Recommendation

This theme investigated news recommendation, with a particular focus on evaluation. All the works are based on our participation in CLEF NEWSREEL News Recommendation Evaluation Lab. We investigated the role of geography in news consumption to understand the relationship between the geographical focus of a news item and the geographical location of news readers followed by the incorporation of geographical information into online deployments of algorithms. After that, we focused on news evaluation, investigating it from several angles. We conducted A/A tests, offline evaluations, online evaluations, and comparisons of algorithm performances across years. This research can be extended in several directions. A very important one is:

Repeating the experiments on other datasets, platforms and settings:
to find out what holds across different datasets, platforms and settings and to confirm or disprove the findings reported in this thesis, repeating the investigation on different datasets, platforms and settings is important. This can help in differentiating dataset and platform effects from the essence of the experiments and the true results they reveal.

11.2.3. Measuring Recommender System Personalization

In this theme, we proposed a user-centric metric of personalization that, by using the recommendation lists and resulting "reaction" lists that users choose to accept, can measure the degree of users' tendency to agree, to converge or to diverge from the differentiation imposed by a recommender system. The proposed metric has been tested on a news recommendation dataset. Further application of the metric on other datasets such as product recommendations and comparison with user studies is needed to further understand what the metric does, and how it concurs or diverges with users' expressed interests. Another possibility to extend is to investigate how the metric is related to or tells us about recommendation diversity, filter bubbles, echo chambers and fairness concerns.

References

References

[1] N. Helberger, *On the Democratic Role of News Recommenders,* Digital Journalism **7**, 993 (2019).

[2] L. Heitz, J. A. Lischka, A. Birrer, B. Paudel, S. Tolmeijer, L. Laugwitz, and A. Bernstein, *Benefits of Diverse News Recommendations for Democracy: A User Study,* Digital Journalism **10**, 1710 (2022).

[3] E. Nechushtai and S. C. Lewis, *What kind of news gatekeepers do we want machines to be? filter bubbles, fragmentation, and the normative dimensions of algorithmic recommendations,* Computers in Human Behavior **90**, 298 (2019).

[4] B. Fetahu, A. Anand, and A. Anand, *How much is wikipedia lagging behind news?* in *Proceedings of the ACM Web Science Conference*, WebSci '15 (ACM, New York, NY, USA, 2015) pp. 28:1–28:9.

[5] J. R. Frank, M. Kleiman-Weiner, D. A. Roberts, F. Niu, C. Zhang, C. Ré, and I. Soboroff, *Building an Entity-Centric Stream Filtering Test Collection for TREC 2012*, Tech. Rep. (Massachusetts Institute of Technology, 2012).

[6] F. Hopfgartner, T. Brodt, J. Seiler, B. Kille, A. Lommatzsch, M. Larson, R. Turrin, and A. Serény, *Benchmarking News Recommendations: The CLEF NewsREEL Use Case,* SIGIR Forum **49**, 129 (2016).

[7] B. Kille, A. Lommatzsch, G. G. Gebremeskel, F. Hopfgartner, M. Larson, J. Seiler, D. Malagoli, A. Serény, T. Brodt, and A. P. de Vries, *Overview of NewsREEL'16: Multi-dimensional Evaluation of Real-time Stream-Recommendation Algorithms,* in *Experimental IR Meets Multilinguality, Multimodality, and Interaction*, edited by N. Fuhr, P. Quaresma, T. Gonçalves, B. Larsen, K. Balog, C. Macdonald, L. Cappellato, and N. Ferro (Springer International Publishing, Cham, 2016) pp. 311–331.

[8] T.-P. Liang, H.-J. Lai, and Y.-C. Ku, *Personalized content recommendation and user satisfaction: Theoretical synthesis and empirical findings,* Journal of Management Information Systems **23**, 45 (2006).

[9] F. Ricci, L. Rokach, and B. Shapira, *Recommender systems: Introduction and challenges,* in *Recommender Systems Handbook*, edited by F. Ricci, L. Rokach, and B. Shapira (Springer US, Boston, MA, 2015) pp. 1–34.

[10] J. R. Frank, S. J. Bauer, M. Kleiman-Weiner, D. A. Roberts, N. Tripuraneni, C. Zhang, C. Re, E. Voorhees, and I. Soboroff, *Evaluating Stream Filtering for Entity Profile*

Updates for TREC 2013 (KBA Track Overview), Tech. Rep. (Massachusetts Inst of Tech Cambridge, 2013).

[11] Y. Koren and R. Bell, *Advances in collaborative filtering,* in *Recommender Systems Handbook*, edited by F. Ricci, L. Rokach, and B. Shapira (Springer US, Boston, MA, 2015) pp. 77–118.

[12] A. Lommatzsch, B. Kille, F. Hopfgartner, and L. Ramming, *Newsreel multimedia at mediaeval 2018: News recommendation with image and text content,* in *Working Notes Proceedings of the MediaEval 2018 Workshop* (CEUR-WS, 2018).

[13] E. Pariser, *The filter bubble: How the new personalized web is changing what we read and how we think* (Penguin, 2011).

[14] K. Balog, *Populating Knowledge Bases,* in *Entity-Oriented Search* (Springer International Publishing, Cham, 2018) pp. 189–222.

[15] *HLT '11: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1* (Association for Computational Linguistics, Stroudsburg, PA, USA, 2011).

[16] NIST, *TAC Knowledge Base Population (KBP),* (2018).

[17] C. Wu, F. Wu, Y. Huang, and X. Xie, *Personalized news recommendation: Methods and challenges,* ACM Trans. Inf. Syst. (2022), 10.1145/3530257, just Accepted.

[18] S. Raza and C. Ding, *News recommender system: A review of recent progress, challenges, and opportunities,* Artif. Intell. Rev. **55**, 749–800 (2022).

[19] I. Soboroff, S. Huang, and D. K. Harman, *TREC 2018 News Track Overview,* in *Text Retrieval Conference* (2018).

[20] B. Kille, A. Lommatzsch, R. Turrin, A. Serény, M. A. Larson, T. Brodt, J. Seiler, and F. Hopfgartner, *Overview of CLEF newsreel 2015: News recommendation evaluation lab,* in *Working Notes of CLEF 2015 - Conference and Labs of the Evaluation forum, Toulouse, France, September 8-11, 2015*, CEUR Workshop Proceedings, Vol. 1391, edited by L. Cappellato, N. Ferro, G. J. F. Jones, and E. SanJuan (CEUR-WS.org, 2015).

[21] T. Brodt and F. Hopfgartner, *Shedding Light on a Living Lab: The CLEF NEWSREEL Open Recommendation Platform,* in *Proceedings of the 5th Information Interaction in Context Symposium*, IIiX '14 (ACM, New York, NY, USA, 2014) pp. 223–226.

[22] D. Jannach, M. Zanker, A. Felfernig, and G. Friedrich, *Recommender Systems: An Introduction*, 1st ed. (Cambridge University Press, New York, NY, USA, 2010).

[23] S. Vrijenhoek, M. Kaya, N. Metoui, J. Möller, D. Odijk, and N. Helberger, *Recommenders with a mission: Assessing diversity in news recommendations,* in *Proceedings of the 2021 Conference on Human Information Interaction and Retrieval*, CHIIR '21 (Association for Computing Machinery, New York, NY, USA, 2021) p. 173–183.

[24] S. Flaxman, S. Goel, and J. M. Rao, *Filter bubbles, echo chambers, and online news consumption,* Public Opinion Quarterly **80**, 298 (2016).

[25] T. T. Nguyen, P.-M. Hui, F. M. Harper, L. Terveen, and J. A. Konstan, *Exploring the filter bubble: the effect of using recommender systems on content diversity,* in *Proceedings of the 23rd international conference on World wide web* (ACM, 2014) pp. 677–686.

[26] V. I. Spitkovsky and A. X. Chang, *A cross-lingual dictionary for English Wikipedia concepts,* in *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, edited by N. Calzolari, K. Choukri, T. Declerck, M. U. Doğan, B. Maegaard, J. Mariani, A. Moreno, J. Odijk, and S. Piperidis (European Language Resources Association (ELRA), Istanbul, Turkey, 2012) pp. 3168–3175.

[27] S. Araujo, G. Gebremeskel, J. He, C. Bosscarino, and A. P. de Vries, *CWI at TREC 2012, KBA Track and Session Track,* in *Proceedings of the 21st Text REtrieval Conference, TREC*, Vol. 12 (2013).

[28] X. Liu and H. Fang, *Leveraging related entities for knowledge base acceleration,* in *Proceedings of the 4th International Workshop on Web-Scale Knowledge Representation Retrieval and Reasoning*, Web-KR '13 (Association for Computing Machinery, New York, NY, USA, 2013) p. 1–4.

[29] K. Balog, H. Ramampiaro, N. Takhirov, and K. Nørvåg, *Multi-step classification approaches to cumulative citation recommendation,* in *Proceedings of the 10th Conference on Open Research Areas in Information Retrieval*, OAIR '13 (Le Centre de hautes études internationales d'informatique documentaire, Paris, FRA, 2013) p. 121–128.

[30] K. Balog and H. Ramampiaro, *Cumulative citation recommendation: classification vs. ranking,* in *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '13 (Association for Computing Machinery, New York, NY, USA, 2013) p. 941–944.

[31] S. Singh, A. Subramanya, F. Pereira, and A. McCallum, *Wikilinks: A large-scale cross-document coreference corpus labeled via links to wikipedia,* University of Massachusetts, Amherst, Tech. Rep. UM-CS-2012 **15** (2012).

[32] S. E. Robertson and I. Soboroff, *The TREC 2002 Filtering Track Report,* in *Proceedings of The Eleventh Text REtrieval Conference, TREC 2002, Gaithersburg, Maryland, USA, November 19-22, 2002*, NIST Special Publication, Vol. 500-251, edited by E. M. Voorhees and L. P. Buckland (National Institute of Standards and Technology (NIST), 2002).

[33] J. Dalton and L. Dietz, *A Neighborhood Relevance Model for Entity Linking,* in *Proceedings of the 10th Conference on Open Research Areas in Information Retrieval* (2013) pp. 149–156.

[34] M. Dredze, P. McNamee, D. Rao, A. Gerber, and T. Finin, *Entity disambiguation for knowledge base population,* in *Proceedings of the 23rd International Conference on Computational Linguistics* (2010) pp. 277–285.

[35] H. Ji and R. Grishman, *Knowledge Base Bopulation: Successful Approaches and Challenges,* in *Proceedings of the 49th Annual Meeting of ACL: Human Language Technologies* (2011) pp. 1148–1158.

[36] J. Wang, D. Song, C.-Y. Lin, and L. Liao, *BIT and MSRA at TREC KBA Track 2013,* in *TREC 2013* (2013).

[37] L. Dietz and J. Dalton, *Umass at TREC 2013 Knowledge Base Acceleration Track,* in *TREC 2013* (2013).

[38] X. Liu and H. Fang, *A Related Entity Based Approach for Knowledge Base Acceleration,* in *TREC 2013* (2013).

[39] V. Bouvier and P. Bellot, *Filtering Entity Centric Documents Using Numerics and Temporals Features within RF Classifier,* in *TREC 2013* (2013).

[40] M. S. Nia, C. Grant, Y. Peng, D. Z. Wang, and M. Petrovic, *University of Florida Knowledge Base Acceleration,* in *TREC 2013* (2013).

[41] M. Efron, C. Willis, P. Organisciak, B. Balsamo, and A. Lucic, *The University of Illinois' Graduate School of LIS at TREC 2013,* in *TREC 2013* (2013).

[42] G. G. Gebremeskel, J. He, A. P. de Vries, and J. Lin, *Cumulative Citation Recommendation: A Feature-aware Comparisons of Approaches,* in *Database and Expert Systems Applications (DEXA)* (IEEE, 2014) pp. 193–197.

[43] G. Baruah, A. Roegiest, and M. D. Smucker, *The Effect of Expanding Relevance Judgements with Duplicates,* in *SIGIR '14 Proceedings of the 37th International ACM SIGIR conference on Research & Development in Information Retrieval* (2014) pp. 1159–1162.

[44] J. R. Frank, M. Kleiman-Weiner, D. A. Roberts, E. Voorhees, and I. Soboroff, *Evaluating Stream Filtering for Entity Profile Updates in TREC 2012, 2013, and 2014 (KBA Track Overview, Notebook Paper)*, Tech. Rep. (Massachusetts Institute of Technology Cambridge, 2014).

[45] G. H. Yang and I. Soboroff, *TREC 2016 Dynamic Domain Track Overview,* in *TREC* (2016).

[46] R. Reinanda, E. Meij, and M. de Rijke, *Document Filtering for Long-tail Entities,* in *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, CIKM '16 (ACM, New York, NY, USA, 2016) pp. 771–780.

[47] D. Graus, D. Odijk, and M. de Rijke, *The birth of collective memories: Analyzing emerging entities in text streams,* Journal of the Association for Information Science and Technology **69**, 773 (2018), https://asistdl.onlinelibrary.wiley.com/doi/pdf/10.1002/asi.24004 .

[48] J. Hoffart, Y. Altun, and G. Weikum, *Discovering Emerging Entities with Ambiguous Names,* in *Proceedings of the 23rd International Conference on World Wide Web*, WWW '14 (ACM, New York, NY, USA, 2014) pp. 385–396.

[49] L. Ma, D. Song, L. Liao, and J. Wang, *A Hybrid Discriminative Mixture Model for Cumulative Citation Recommendation,* IEEE Transactions on Knowledge and Data Engineering (2019).

[50] T. Mikolov, K. Chen, G. Corrado, and J. Dean, *Efficient Estimation of Word Representations in Vector Space,* in *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*, edited by Y. Bengio and Y. LeCun (2013).

[51] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, *Distributed Representations of Words and Phrases and their Compositionality,* in *Advances in Neural Information Processing Systems*, Vol. 26, edited by C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Weinberger (Curran Associates, Inc., 2013).

[52] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, J. Klingner, A. Shah, M. Johnson, X. Liu, L. Kaiser, S. Gouws, Y. Kato, T. Kudo, H. Kazawa, K. Stevens, G. Kurian, N. Patil, W. Wang, C. Young, J. Smith, J. Riesa, A. Rudnick, O. Vinyals, G. Corrado, M. Hughes, and J. Dean, *Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation,* CoRR **abs/1609.08144** (2016), arXiv:1609.08144 .

[53] T. Kudo and J. Richardson, *SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing,* in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, edited by E. Blanco and W. Lu (Association for Computational Linguistics, Brussels, Belgium, 2018) pp. 66–71.

[54] Y. Kim, C. Denton, L. Hoang, and A. M. Rush, *Structured Attention Networks,* in *International Conference on Learning Representations* (2017).

[55] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, *Attention is All you Need,* in *Advances in Neural Information Processing Systems 30*, edited by I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Curran Associates, Inc., 2017) pp. 5998–6008.

[56] M. Johnson, M. Schuster, Q. V. Le, M. Krikun, Y. Wu, Z. Chen, N. Thorat, F. Viégas, M. Wattenberg, G. Corrado, M. Hughes, and J. Dean, *Google's multilingual neural machine translation system: Enabling zero-shot translation,* Transactions of the Association for Computational Linguistics **5**, 339 (2017).

[57] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, *Deep contextualized word representations,* in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, edited by M. Walker, H. Ji, and A. Stent (Association for Computational Linguistics, New Orleans, Louisiana, 2018) pp. 2227–2237.

[58] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, *BERT: Pre-training of deep bidirectional transformers for language understanding,* in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, edited by J. Burstein, C. Doran, and T. Solorio (Association for Computational Linguistics, Minneapolis, Minnesota, 2019) pp. 4171–4186.

[59] O.-E. Ganea and T. Hofmann, *Deep joint entity disambiguation with local neural attention,* in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, edited by M. Palmer, R. Hwa, and S. Riedel (Association for Computational Linguistics, Copenhagen, Denmark, 2017) pp. 2619–2629.

[60] J. Raiman and O. Raiman, *DeepType: multilingual entity linking by neural type system evolution,* in *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence*, AAAI'18/IAAI'18/EAAI'18 (AAAI Press, 2018).

[61] N. Kolitsas, O. Ganea, and T. Hofmann, *End-to-End Neural Entity Linking,* CoRR **abs/1808.07699** (2018), arXiv:1808.07699 .

[62] P. H. Martins, Z. Marinho, and A. F. T. Martins, *Joint learning of named entity recognition and entity linking,* in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, edited by F. Alva-Manchego, E. Choi, and D. Khashabi (Association for Computational Linguistics, Florence, Italy, 2019) pp. 190–196.

[63] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, *Language models are unsupervised multitask learners,* OpenAI Blog **1** (2019).

[64] F. Petroni, T. Rocktäschel, S. Riedel, P. Lewis, A. Bakhtin, Y. Wu, and A. Miller, *Language models as knowledge bases?* in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, edited by K. Inui, J. Jiang, V. Ng, and X. Wan (Association for Computational Linguistics, Hong Kong, China, 2019) pp. 2463–2473.

[65] N. Noy, Y. Gao, A. Jain, A. Narayanan, A. Patterson, and J. Taylor, *Industry-scale Knowledge Graphs: Lessons and Challenges,* Queue **17**, 20:48 (2019).

[66] C. Dogan, A. Dutra, A. Gara, A. Gemma, L. Shi, M. Sigamani, and E. Walters, *Fine-Grained Named Entity Recognition using ELMo and Wikidata,* CoRR **abs/1904.10503** (2019), arXiv:1904.10503 .

[67] F. Souza, R. Nogueira, and R. Lotufo, *Bertimbau: Pretrained bert models for brazilian portuguese,* in *Intelligent Systems*, edited by R. Cerri and R. C. Prati (Springer International Publishing, Cham, 2020) pp. 403–417.

[68] M. Chen, Z. Chu, Y. Chen, K. Stratos, and K. Gimpel, *EntEval: A Holistic Evaluation Benchmark for Entity Representations,* in *Proc. of EMNLP* (2019).

[69] L. Logeswaran, M.-W. Chang, K. Lee, K. Toutanova, J. Devlin, and H. Lee, *Zero-Shot Entity Linking by Reading Entity Descriptions,* in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics* (2019) pp. 3449–3460.

[70] E. F. Tjong Kim Sang and F. De Meulder, *Introduction to the CoNLL-2003 Shared Task: Language-independent Named Entity Recognition,* in *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003 - Volume 4*, CONLL '03 (Association for Computational Linguistics, Stroudsburg, PA, USA, 2003) pp. 142–147.

[71] J. Hoffart, M. A. Yosef, I. Bordino, H. Fürstenau, M. Pinkal, M. Spaniol, B. Taneva, S. Thater, and G. Weikum, *Robust Disambiguation of Named Entities in Text,* in *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing* (Association for Computational Linguistics, Edinburgh, Scotland, UK., 2011) pp. 782–792.

[72] M. F. Dacrema, P. Cremonesi, and D. Jannach, *Are We Really Making Much Progress? A Worrying Analysis of Recent Neural Recommendation Approaches,* in *Proceedings of the 13th ACM Conference on Recommender Systems*, RecSys '19 (ACM, New York, NY, USA, 2019) pp. 101–109.

[73] A. Akbik, T. Bergmann, D. Blythe, K. Rasul, S. Schweter, and R. Vollgraf, *FLAIR: An easy-to-use framework for state-of-the-art NLP,* in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, edited by W. Ammar, A. Louis, and N. Mostafazadeh (Association for Computational Linguistics, Minneapolis, Minnesota, 2019) pp. 54–59.

[74] T. Ayoola, S. Tyagi, J. Fisher, C. Christodoulopoulos, and A. Pierleoni, *ReFinED: An efficient zero-shot-capable approach to end-to-end entity linking,* in *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Industry Track*, edited by A. Loukina, R. Gangadharaiah, and B. Min (Association for Computational Linguistics, Hybrid: Seattle, Washington + Online, 2022) pp. 209–220.

[75] J. M. van Hulst, F. Hasibi, K. Dercksen, K. Balog, and A. P. de Vries, *REL: An Entity Linker Standing on the Shoulders of Giants,* in *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '20 (Association for Computing Machinery, New York, NY, USA, 2020) p. 2197–2200.

[76] N. De Cao, G. Izacard, S. Riedel, and F. Petroni, *Autoregressive Entity Retrieval,* in *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021* (OpenReview.net, 2021).

[77] L. Wu, F. Petroni, M. Josifoski, S. Riedel, and L. Zettlemoyer, *Scalable zero-shot entity linking with dense entity retrieval,* in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, edited by B. Webber, T. Cohn, Y. He, and Y. Liu (Association for Computational Linguistics, Online, 2020) pp. 6397–6407.

[78] H. Peng, K.-W. Chang, and D. Roth, *A Joint Framework for Coreference Resolution and Mention Head Detection,* in *Proceedings of the Nineteenth Conference on Computational Natural Language Learning* (Association for Computational Linguistics, Beijing, China, 2015) pp. 12–21.

[79] L. Wu, I. E.-H. Yen, K. Xu, F. Xu, A. Balakrishnan, P.-Y. Chen, P. Ravikumar, and M. J. Witbrock, *Word mover's embedding: From Word2Vec to document embedding,* in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, edited by E. Riloff, D. Chiang, J. Hockenmaier, and J. Tsujii (Association for Computational Linguistics, Brussels, Belgium, 2018) pp. 4524–4534.

[80] M. Guo, Y. Yang, K. Stevens, D. Cer, H. Ge, Y.-h. Sung, B. Strope, and R. Kurzweil, *Hierarchical document encoder for parallel corpus mining,* in *Proceedings of the Fourth Conference on Machine Translation (Volume 1: Research Papers)*, edited by O. Bojar, R. Chatterjee, C. Federmann, M. Fishel, Y. Graham, B. Haddow, M. Huck, A. J. Yepes, P. Koehn, A. Martins, C. Monz, M. Negri, A. Névéol, M. Neves, M. Post, M. Turchi, and K. Verspoor (Association for Computational Linguistics, Florence, Italy, 2019) pp. 64–72.

[81] J. Guo, Y. Fan, L. Pang, L. Yang, Q. Ai, H. Zamani, C. Wu, W. B. Croft, and X. Cheng, *A Deep Look into neural ranking models for information retrieval,* Information Processing & Management **57**, 102067 (2020).

[82] S. Verberne, *Pretrained Transformers for Text Ranking: BERT and Beyond,* Computational Linguistics **49**, 253 (2023), https://direct.mit.edu/coli/article-pdf/49/1/253/2068903/coli_r_00468.pdf .

[83] Z. Dai and J. Callan, *Deeper Text Understanding for IR with Contextual Neural Language Modeling,* in *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2019, Paris, France, July 21-25, 2019*, edited by B. Piwowarski, M. Chevalier, É. Gaussier, Y. Maarek, J. Nie, and F. Scholer (ACM, 2019) pp. 985–988.

[84] W. Yang, H. Zhang, and J. Lin, *Simple Applications of BERT for Ad Hoc Document Retrieval,* CoRR **abs/1903.10972** (2019), arXiv:1903.10972 .

[85] L. Dietz, H. Bast, S. Chatterjee, J. Dalton, J.-Y. Nie, and R. Nogueira, *Neuro-Symbolic Representations for Information Retrieval,* in *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information*

Retrieval, SIGIR '23 (Association for Computing Machinery, New York, NY, USA, 2023) p. 3436–3439.

[86] C. Kamphuis, A. Lin, S. Yang, J. Lin, A. P. de Vries, and F. Hasibi, *MMEAD: MS MARCO Entity Annotations and Disambiguations,* in *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '23 (Association for Computing Machinery, New York, NY, USA, 2023) p. 2817–2825.

[87] P. Gage, *A New Algorithm for Sata Compression,* The C Users Journal **12**, 23–38 (1994).

[88] N. Kandpal, H. Deng, A. Roberts, E. Wallace, and C. Raffel, *Large Language Models Struggle to Learn Long-Tail Knowledge,* in *Proceedings of the 40th International Conference on Machine Learning*, ICML'23 (JMLR.org, 2023).

[89] A. Mallen, A. Asai, V. Zhong, R. Das, D. Khashabi, and H. Hajishirzi, *When not to trust language models: Investigating effectiveness of parametric and non-parametric memories,* in *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, edited by A. Rogers, J. Boyd-Graber, and N. Okazaki (Association for Computational Linguistics, Toronto, Canada, 2023) pp. 9802–9822.

[90] K. Spärck Jones, *IDF term weighting and IR research lessons,* Journal of Documentation **60**, 521 (2004).

[91] J. Kasai, K. Sakaguchi, Y. Takahashi, R. L. Bras, A. Asai, X. Yu, D. Radev, N. A. Smith, Y. Choi, and K. Inui, *RealTime QA: What's the Answer Right Now?* (2022), arXiv:2207.13332 [cs.CL] .

[92] G. Izacard, P. Lewis, M. Lomeli, L. Hosseini, F. Petroni, T. Schick, J. Dwivedi-Yu, A. Joulin, S. Riedel, and E. Grave, *Atlas: Few-shot Learning with Retrieval Augmented Language Models,* (2022), arXiv:2208.03299 [cs.CL] .

[93] L. Xue, A. Barua, N. Constant, R. Al-Rfou, S. Narang, M. Kale, A. Roberts, and C. Raffel, *ByT5: Towards a token-free future with pre-trained byte-to-byte models,* Transactions of the Association for Computational Linguistics **10**, 291 (2022).

[94] E. Gabrilovich, S. Dumais, and E. Horvitz, *Newsjunkie: providing personalized newsfeeds via analysis of information novelty,* in *Proceedings of the 13th international conference on World Wide Web* (ACM, 2004) pp. 482–490.

[95] L. Li, D. Wang, T. Li, D. Knox, and B. Padmanabhan, *Scene: a scalable two-stage personalized news recommendation system,* in *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval* (ACM, 2011) pp. 125–134.

[96] C. Kulkarni and E. Chi, *All the news that's fit to read: a study of social annotations for news reading,* in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (ACM, 2013) pp. 2407–2416.

[97] J. Jancsary, F. Neubarth, and H. Trost, *Towards context-aware personalization and a broad perspective on the semantics of news articles,* in *Proceedings of the fourth ACM conference on Recommender systems* (ACM, 2010) pp. 289–292.

[98] A. Said, J. Lin, A. Bellogín, and A. P. de Vries, *A month in the life of a production news recommender system,* in *Proceedings of the 2013 workshop on Living labs for information retrieval evaluation* (ACM, 2013) pp. 7–10.

[99] G. G. Gebremeskel and A. P. de Vries, *The role of geographic information in news consumption,* in *Proceedings of the 24th International Conference on World Wide Web Companion* (International World Wide Web Conferences Steering Committee, 2015).

[100] F. Hopfgartner, B. Kille, A. Lommatzsch, T. Plumbaum, T. Brodt, and T. Heintz, *Benchmarking news recommendations in a living lab,* in *Information Access Evaluation. Multilinguality, Multimodality, and Interaction* (Springer, 2014) pp. 250–267.

[101] F. Hopfgartner, T. Brodt, J. Seiler, B. Kille, A. Lommatzsch, M. Larson, R. Turrin, and A. Serény, *Benchmarking News Recommendations: The CLEF NewsREEL Use Case,* SIGIR Forum **49**, 129–136 (2016).

[102] T. Brodt and F. Hopfgartner, *Shedding Light on a Living Lab: The CLEF NEWSREEL Open Recommendation Platform,* in *Proceedings of the 5th Information Interaction in Context Symposium*, IIiX '14 (Association for Computing Machinery, New York, NY, USA, 2014) p. 223–226.

[103] F. Garcin, B. Faltings, O. Donatsch, A. Alazzawi, C. Bruttin, and A. Huber, *Offline and online evaluation of news recommender systems at swissinfo. ch,* in *Proceedings of the 8th ACM Conference on Recommender systems* (ACM, 2014) pp. 169–176.

[104] J. Beel, M. Genzmehr, S. Langer, A. Nürnberger, and B. Gipp, *A comparative analysis of offline and online evaluations and discussion of research paper recommender system evaluation,* in *Proceedings of the International Workshop on Reproducibility and Replication in Recommender Systems Evaluation* (ACM, 2013) pp. 7–14.

[105] E. Kirshenbaum, G. Forman, and M. Dugan, *A live comparison of methods for personalized article recommendation at forbes. com,* in *Machine Learning and Knowledge Discovery in Databases* (Springer, 2012) pp. 51–66.

[106] S. M. McNee, N. Kapoor, and J. A. Konstan, *Don't look stupid: avoiding pitfalls when recommending research papers,* in *Proceedings of the 2006 20th anniversary conference on Computer supported cooperative work* (ACM, 2006) pp. 171–180.

[107] G. G. Gebremeskel and A. P. de Vries, *Random performance differences between online recommender system algorithms,* in *Experimental IR Meets Multilinguality, Multimodality, and Interaction 7th International Conference of the CLEF Association, CLEF 2016, Évora, Portugal, September 5-8, 2016.*, edited by N. Fuhr, P. Quaresma, B. Larsen, T. Goncalves, K. Balog, C. Macdonald, L. Cappellato, and N. Ferro (Springer, 2016).

[108] B. Kille, A. Lommatzsch, R. Turrin, A. Serény, M. Larson, T. Brodt, J. Seiler, and F. Hopfgartner, *Stream-based recommendations: Online and offline evaluation as a service,* in *International Conference of the Cross-Language Evaluation Forum for European Languages* (Springer, 2015) pp. 497–517.

[109] A. Lommatzsch, B. Kille, F. Hopfgartner, M. Larson, T. Brodt, J. Seiler, and Ö. Özgöbek, *CLEF 2017 NewsREEL Overview: A Stream-Based Recommender Task for Evaluation and Education,* in *Experimental IR Meets Multilinguality, Multimodality, and Interaction*, edited by G. J. Jones, S. Lawless, J. Gonzalo, L. Kelly, L. Goeuriot, T. Mandl, L. Cappellato, and N. Ferro (Springer International Publishing, Cham, 2017) pp. 239–254.

[110] B. Kille, A. Lommatzsch, F. Hopfgartner, M. Larson, J. Seiler, D. Malagoli, A. Serény, and T. Brodt, *CLEF NewsREEL 2016: Comparing multi-dimensional offline and online evaluation of news recommender systems,* (CEUR workshop proceedings, 2016).

[111] F. Hopfgartner, K. Balog, A. Lommatzsch, L. Kelly, B. Kille, A. Schuth, and M. Larson, *Continuous evaluation of large-scale information access systems: a case for living labs,* in *Information Retrieval Evaluation in a Changing World* (Springer, 2019) pp. 511–543.

[112] M. Ge, C. Delgado-Battenfeld, and D. Jannach, *Beyond Accuracy: Evaluating Recommender Systems by Coverage and Serendipity,* in *Proceedings of the Fourth ACM Conference on Recommender Systems*, RecSys '10 (ACM, New York, NY, USA, 2010) pp. 257–260.

[113] S. Okura, Y. Tagami, S. Ono, and A. Tajima, *Embedding-based News Recommendation for Millions of Users,* in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '17 (ACM, New York, NY, USA, 2017) pp. 1933–1942.

[114] C. Wu, F. Wu, M. An, J. Huang, Y. Huang, and X. Xie, *NPA: Neural News Recommendation with Personalized Attention,* in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '19 (ACM, New York, NY, USA, 2019) pp. 2576–2584.

[115] V. Kumar, D. Khattar, S. Gupta, and V. Varma, *Word Semantics Based 3-D Convolutional Neural Networks for News Recommendation,* in *2017 IEEE International Conference on Data Mining Workshops (ICDMW)* (IEEE, 2017) pp. 761–764.

[116] G. Zheng, F. Zhang, Z. Zheng, Y. Xiang, N. J. Yuan, X. Xie, and Z. Li, *DRN: A Deep Reinforcement Learning Framework for News Recommendation,* in *Proceedings of the 2018 World Wide Web Conference*, WWW '18 (International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, Switzerland, 2018) pp. 167–176.

[117] H. Wang, F. Zhang, X. Xie, and M. Guo, *DKN: Deep knowledge-aware network for news recommendation,* in *Proceedings of the 2018 World Wide Web Conference*

(International World Wide Web Conferences Steering Committee, 2018) pp. 1835–1844.

[118] C. Wu, F. Wu, M. An, J. Huang, Y. Huang, and X. Xie, *Neural news recommendation with attentive multi-view learning,* in *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, IJCAI'19 (AAAI Press, 2019) p. 3863–3869.

[119] V. Kumar, D. Khattar, S. Gupta, M. Gupta, and V. Varma, *Deep Neural Architecture for News Recommendation,* in *CLEF (Working Notes)* (2017).

[120] R. Cañamares and P. Castells, *On target item sampling in offline recommender system evaluation,* in *Fourteenth ACM Conference on Recommender Systems*, RecSys '20 (Association for Computing Machinery, New York, NY, USA, 2020) p. 259–268.

[121] N. Fuhr, *Reproducibility and Validity in CLEF,* in *Information Retrieval Evaluation in a Changing World: Lessons Learned from 20 Years of CLEF*, edited by N. Ferro and C. Peters (Springer International Publishing, Cham, 2019) pp. 555–564.

[122] N. Ferro, N. Fuhr, K. Järvelin, N. Kando, M. Lippold, and J. Zobel, *Increasing Reproducibility in IR: Findings from the Dagstuhl Seminar on "Reproducibility of Data-Oriented Experiments in e-Science",* SIGIR Forum **50**, 68 (2016).

[123] B. Xiao and I. Benbasat, *E-commerce product recommendation agents: Use, characteristics, and impact,* Mis Quarterly **31**, 137 (2007).

[124] A. Hannak, P. Sapiezynski, A. Molavi Kakhki, B. Krishnamurthy, D. Lazer, A. Mislove, and C. Wilson, *Measuring personalization of web search,* in *Proceedings of the 22nd international conference on World Wide Web* (International World Wide Web Conferences Steering Committee, 2013) pp. 527–538.

[125] P. Pu, L. Chen, and R. Hu, *A user-centric evaluation framework for recommender systems,* in *Proceedings of the fifth ACM conference on Recommender systems* (ACM, 2011) pp. 157–164.

[126] J. Teevan, S. T. Dumais, and E. Horvitz, *Potential for personalization,* ACM Transactions on Computer-Human Interaction (TOCHI) **17**, 4 (2010).

[127] R. K. Garrett, *Echo chambers online?: Politically motivated selective exposure among internet news users,* Journal of Computer-Mediated Communication **14**, 265 (2009).

[128] S. Iyengar and K. S. Hahn, *Red media, blue media: Evidence of ideological selectivity in media use,* Journal of Communication **59**, 19 (2009).

[129] S. A. Munson and P. Resnick, *Presenting diverse political opinions: how and how much,* in *Proceedings of the SIGCHI conference on human factors in computing systems* (ACM, 2010) pp. 1457–1466.

[130] Q. V. Liao and W.-T. Fu, *Beyond the filter bubble: interactive effects of perceived threat and topic involvement on selective exposure to information,* in *Proceedings of the SIGCHI conference on human factors in computing systems* (ACM, 2013) pp. 2359–2368.

[131] X. Yi, L. Hong, E. Zhong, N. Liu, and S. Rajan, *Beyond clicks: Dwell time for personalization,* in *RecSys'14* (2014).

[132] K. Järvelin and J. Kekäläinen, *Cumulated gain-based evaluation of ir techniques,* ACM Trans. Inf. Syst. **20**, 422 (2002).

[133] R. Mehrotra, A. Anderson, F. Diaz, A. Sharma, H. Wallach, and E. Yilmaz, *Auditing search engines for differential satisfaction across demographics,* in *Proceedings of the 26th International Conference on World Wide Web Companion* (International World Wide Web Conferences Steering Committee, 2017) pp. 626–633.

[134] S. Goel, J. M. Hofman, and M. I. Sirer, *Who does what on the Web: A large-scale study of browsing behavior.* in *ICWSM* (2012).

[135] A. Bellogín, P. Castells, and I. Cantador, *Precision-oriented evaluation of recommender systems: An algorithmic comparison,* in *Proceedings of the Fifth ACM Conference on Recommender Systems*, RecSys '11 (ACM, New York, NY, USA, 2011) pp. 333–336.

[136] M. D. Ekstrand, M. Tian, I. M. Azpiazu, J. D. Ekstrand, O. Anuyah, D. McNeill, and M. S. Pera, *All the cool kids, how do they fit in?: Popularity and demographic biases in recommender evaluation and effectiveness,* in *Proceedings of the 1st Conference on Fairness, Accountability and Transparency*, Proceedings of Machine Learning Research, Vol. 81, edited by S. A. Friedler and C. Wilson (PMLR, New York, NY, USA, 2018) pp. 172–186.

[137] R. Cañamares and P. Castells, *Should I follow the crowd?: A probabilistic analysis of the effectiveness of popularity in recommender systems,* in *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, SIGIR '18 (ACM, New York, NY, USA, 2018) pp. 415–424.

[138] A. Bellogín, P. Castells, and I. Cantador, *Statistical biases in information retrieval metrics for recommender systems,* Information Retrieval Journal **20**, 606 (2017).

[139] D. Jannach, L. Lerche, I. Kamehkhosh, and M. Jugovac, *What recommenders recommend: An analysis of recommendation biases and possible countermeasures,* User Modeling and User-Adapted Interaction **25**, 427 (2015).

[140] H. Steck, *Item popularity and recommendation accuracy,* in *Proceedings of the Fifth ACM Conference on Recommender Systems*, RecSys '11 (ACM, New York, NY, USA, 2011) pp. 125–132.

[141] N. Helberger, *On the democratic role of news recommenders,* Digital Journalism **0**, 1 (2019), https://doi.org/10.1080/21670811.2019.1623700 .

[142] L. Sweeney, *Discrimination in online ad delivery,* Queue **11**, 10:10 (2013).

[143] C. Castillo, *Fairness and transparency in ranking,* SIGIR Forum **52**, 64 (2019).

[144] R. Baeza-Yates, *Bias on the Web,* Commun. ACM **61**, 54 (2018).

[145] B. Edizel, F. Bonchi, S. Hajian, A. Panisson, and T. Tassa, *Fairecsys: mitigating algorithmic bias in recommender systems,* International Journal of Data Science and Analytics (2019), 10.1007/s41060-019-00181-5.

[146] R. Burke, *Multisided fairness for recommendation,* CoRR **abs/1707.00093** (2017), arXiv:1707.00093 .

[147] R. Binns, *Fairness in machine learning: Lessons from political philosophy,* Proceedings of Machine Learning Research **81**, 1 (2017).

[148] A. Bellogín, G. G. Gebremeskel, J. He, J. Lin, A. Said, T. Samar, A. P. de Vries, and J. B. Vuurens, *CWI and TU Delft at TREC 2013: Contextual Suggestion, Federated Web Search, KBA, and Web Tracks,* in *Proceedings of the TExt Retrieval Conference (TREC)* (2013).

[149] G. G. Gebremeskel and A. P. de Vries, *Entity-centric stream filtering and ranking: Filtering and unfilterable documents.* in *ECIR* (2015) pp. 303–314.

[150] G. G. Gebremeskel and A. P. de Vries, *The degree of randomness in a live recommender systems evaluation.* in *CLEF (Working Notes)* (2015) p. 41.

[151] G. G. Gebremeskel and A. P. de Vries, *Random performance differences between online recommender system algorithms,* in *Experimental IR Meets Multilinguality, Multimodality, and Interaction: 7th International Conference of the CLEF Association, CLEF 2016, Évora, Portugal, September 5-8, 2016, Proceedings 7* (Springer, 2016) pp. 187–198.

[152] G. G. Gebremeskel and A. P. de Vries, *Recommender systems evaluations: Offline, online, time and a/a test,* in *CLEF (Working Notes)* (2016) pp. 642–656.

[153] G. G. Gebremeskel and A. P. de Vries, *Pull–push: a measure of over-or underpersonalization in recommendation,* International Journal of Data Science and Analytics , 1 (2022).

SIKS Dissertatiereeks

2016
01 Syed Saiden Abbas (RUN), Recognition of Shapes by Humans and Machines
02 Michiel Christiaan Meulendijk (UU), Optimizing medication reviews through decision support: prescribing a better pill to swallow
03 Maya Sappelli (RUN), Knowledge Work in Context: User Centered Knowledge Worker Support
04 Laurens Rietveld (VUA), Publishing and Consuming Linked Data
05 Evgeny Sherkhonov (UvA), Expanded Acyclic Queries: Containment and an Application in Explaining Missing Answers
06 Michel Wilson (TUD), Robust scheduling in an uncertain environment
07 Jeroen de Man (VUA), Measuring and modeling negative emotions for virtual training
08 Matje van de Camp (TiU), A Link to the Past: Constructing Historical Social Networks from Unstructured Data
09 Archana Nottamkandath (VUA), Trusting Crowdsourced Information on Cultural Artefacts
10 George Karafotias (VUA), Parameter Control for Evolutionary Algorithms
11 Anne Schuth (UvA), Search Engines that Learn from Their Users
12 Max Knobbout (UU), Logics for Modelling and Verifying Normative Multi-Agent Systems
13 Nana Baah Gyan (VUA), The Web, Speech Technologies and Rural Development in West Africa - An ICT4D Approach
14 Ravi Khadka (UU), Revisiting Legacy Software System Modernization
15 Steffen Michels (RUN), Hybrid Probabilistic Logics - Theoretical Aspects, Algorithms and Experiments
16 Guangliang Li (UvA), Socially Intelligent Autonomous Agents that Learn from Human Reward
17 Berend Weel (VUA), Towards Embodied Evolution of Robot Organisms
18 Albert Meroño Peñuela (VUA), Refining Statistical Data on the Web
19 Julia Efremova (TU/e), Mining Social Structures from Genealogical Data
20 Daan Odijk (UvA), Context & Semantics in News & Web Search
21 Alejandro Moreno Célleri (UT), From Traditional to Interactive Playspaces: Automatic Analysis of Player Behavior in the Interactive Tag Playground
22 Grace Lewis (VUA), Software Architecture Strategies for Cyber-Foraging Systems
23 Fei Cai (UvA), Query Auto Completion in Information Retrieval
24 Brend Wanders (UT), Repurposing and Probabilistic Integration of Data; An Iterative and data model independent approach
25 Julia Kiseleva (TU/e), Using Contextual Information to Understand Searching and Browsing Behavior
26 Dilhan Thilakarathne (VUA), In or Out of Control: Exploring Computational Models to Study the Role of Human Awareness and Control in Behavioural Choices, with Applications in Aviation and Energy Management Domains
27 Wen Li (TUD), Understanding Geo-spatial Information on Social Media
28 Mingxin Zhang (TUD), Large-scale Agent-based Social Simulation - A study on epidemic prediction and control

29 Nicolas Höning (TUD), Peak reduction in decentralised electricity systems - Markets and prices for flexible planning
30 Ruud Mattheij (TiU), The Eyes Have It
31 Mohammad Khelghati (UT), Deep web content monitoring
32 Eelco Vriezekolk (UT), Assessing Telecommunication Service Availability Risks for Crisis Organisations
33 Peter Bloem (UvA), Single Sample Statistics, exercises in learning from just one example
34 Dennis Schunselaar (TU/e), Configurable Process Trees: Elicitation, Analysis, and Enactment
35 Zhaochun Ren (UvA), Monitoring Social Media: Summarization, Classification and Recommendation
36 Daphne Karreman (UT), Beyond R2D2: The design of nonverbal interaction behavior optimized for robot-specific morphologies
37 Giovanni Sileno (UvA), Aligning Law and Action - a conceptual and computational inquiry
38 Andrea Minuto (UT), Materials that Matter - Smart Materials meet Art & Interaction Design
39 Merijn Bruijnes (UT), Believable Suspect Agents; Response and Interpersonal Style Selection for an Artificial Suspect
40 Christian Detweiler (TUD), Accounting for Values in Design
41 Thomas King (TUD), Governing Governance: A Formal Framework for Analysing Institutional Design and Enactment Governance
42 Spyros Martzoukos (UvA), Combinatorial and Compositional Aspects of Bilingual Aligned Corpora
43 Saskia Koldijk (RUN), Context-Aware Support for Stress Self-Management: From Theory to Practice
44 Thibault Sellam (UvA), Automatic Assistants for Database Exploration
45 Bram van de Laar (UT), Experiencing Brain-Computer Interface Control
46 Jorge Gallego Perez (UT), Robots to Make you Happy
47 Christina Weber (UL), Real-time foresight - Preparedness for dynamic innovation networks
48 Tanja Buttler (TUD), Collecting Lessons Learned
49 Gleb Polevoy (TUD), Participation and Interaction in Projects. A Game-Theoretic Analysis
50 Yan Wang (TiU), The Bridge of Dreams: Towards a Method for Operational Performance Alignment in IT-enabled Service Supply Chains

2017 01 Jan-Jaap Oerlemans (UL), Investigating Cybercrime
02 Sjoerd Timmer (UU), Designing and Understanding Forensic Bayesian Networks using Argumentation
03 Daniël Harold Telgen (UU), Grid Manufacturing; A Cyber-Physical Approach with Autonomous Products and Reconfigurable Manufacturing Machines
04 Mrunal Gawade (CWI), Multi-core Parallelism in a Column-store
05 Mahdieh Shadi (UvA), Collaboration Behavior
06 Damir Vandic (EUR), Intelligent Information Systems for Web Product Search
07 Roel Bertens (UU), Insight in Information: from Abstract to Anomaly
08 Rob Konijn (VUA), Detecting Interesting Differences:Data Mining in Health Insurance Data using Outlier Detection and Subgroup Discovery
09 Dong Nguyen (UT), Text as Social and Cultural Data: A Computational Perspective on Variation in Text
10 Robby van Delden (UT), (Steering) Interactive Play Behavior
11 Florian Kunneman (RUN), Modelling patterns of time and emotion in Twitter #anticipointment

12 Sander Leemans (TU/e), Robust Process Mining with Guarantees
13 Gijs Huisman (UT), Social Touch Technology - Extending the reach of social touch through haptic technology
14 Shoshannah Tekofsky (TiU), You Are Who You Play You Are: Modelling Player Traits from Video Game Behavior
15 Peter Berck (RUN), Memory-Based Text Correction
16 Aleksandr Chuklin (UvA), Understanding and Modeling Users of Modern Search Engines
17 Daniel Dimov (UL), Crowdsourced Online Dispute Resolution
18 Ridho Reinanda (UvA), Entity Associations for Search
19 Jeroen Vuurens (UT), Proximity of Terms, Texts and Semantic Vectors in Information Retrieval
20 Mohammadbashir Sedighi (TUD), Fostering Engagement in Knowledge Sharing: The Role of Perceived Benefits, Costs and Visibility
21 Jeroen Linssen (UT), Meta Matters in Interactive Storytelling and Serious Gaming (A Play on Worlds)
22 Sara Magliacane (VUA), Logics for causal inference under uncertainty
23 David Graus (UvA), Entities of Interest — Discovery in Digital Traces
24 Chang Wang (TUD), Use of Affordances for Efficient Robot Learning
25 Veruska Zamborlini (VUA), Knowledge Representation for Clinical Guidelines, with applications to Multimorbidity Analysis and Literature Search
26 Merel Jung (UT), Socially intelligent robots that understand and respond to human touch
27 Michiel Joosse (UT), Investigating Positioning and Gaze Behaviors of Social Robots: People's Preferences, Perceptions and Behaviors
28 John Klein (VUA), Architecture Practices for Complex Contexts
29 Adel Alhuraibi (TiU), From IT-BusinessStrategic Alignment to Performance: A Moderated Mediation Model of Social Innovation, and Enterprise Governance of IT"
30 Wilma Latuny (TiU), The Power of Facial Expressions
31 Ben Ruijl (UL), Advances in computational methods for QFT calculations
32 Thaer Samar (RUN), Access to and Retrievability of Content in Web Archives
33 Brigit van Loggem (OU), Towards a Design Rationale for Software Documentation: A Model of Computer-Mediated Activity
34 Maren Scheffel (OU), The Evaluation Framework for Learning Analytics
35 Martine de Vos (VUA), Interpreting natural science spreadsheets
36 Yuanhao Guo (UL), Shape Analysis for Phenotype Characterisation from High-throughput Imaging
37 Alejandro Montes Garcia (TU/e), WiBAF: A Within Browser Adaptation Framework that Enables Control over Privacy
38 Alex Kayal (TUD), Normative Social Applications
39 Sara Ahmadi (RUN), Exploiting properties of the human auditory system and compressive sensing methods to increase noise robustness in ASR
40 Altaf Hussain Abro (VUA), Steer your Mind: Computational Exploration of Human Control in Relation to Emotions, Desires and Social Support For applications in human-aware support systems
41 Adnan Manzoor (VUA), Minding a Healthy Lifestyle: An Exploration of Mental Processes and a Smart Environment to Provide Support for a Healthy Lifestyle
42 Elena Sokolova (RUN), Causal discovery from mixed and missing data with applications on ADHD datasets
43 Maaike de Boer (RUN), Semantic Mapping in Video Retrieval

	44	Garm Lucassen (UU), Understanding User Stories - Computational Linguistics in Agile Requirements Engineering
	45	Bas Testerink (UU), Decentralized Runtime Norm Enforcement
	46	Jan Schneider (OU), Sensor-based Learning Support
	47	Jie Yang (TUD), Crowd Knowledge Creation Acceleration
	48	Angel Suarez (OU), Collaborative inquiry-based learning
2018	01	Han van der Aa (VUA), Comparing and Aligning Process Representations
	02	Felix Mannhardt (TU/e), Multi-perspective Process Mining
	03	Steven Bosems (UT), Causal Models For Well-Being: Knowledge Modeling, Model-Driven Development of Context-Aware Applications, and Behavior Prediction
	04	Jordan Janeiro (TUD), Flexible Coordination Support for Diagnosis Teams in Data-Centric Engineering Tasks
	05	Hugo Huurdeman (UvA), Supporting the Complex Dynamics of the Information Seeking Process
	06	Dan Ionita (UT), Model-Driven Information Security Risk Assessment of Socio-Technical Systems
	07	Jieting Luo (UU), A formal account of opportunism in multi-agent systems
	08	Rick Smetsers (RUN), Advances in Model Learning for Software Systems
	09	Xu Xie (TUD), Data Assimilation in Discrete Event Simulations
	10	Julienka Mollee (VUA), Moving forward: supporting physical activity behavior change through intelligent technology
	11	Mahdi Sargolzaei (UvA), Enabling Framework for Service-oriented Collaborative Networks
	12	Xixi Lu (TU/e), Using behavioral context in process mining
	13	Seyed Amin Tabatabaei (VUA), Computing a Sustainable Future
	14	Bart Joosten (TiU), Detecting Social Signals with Spatiotemporal Gabor Filters
	15	Naser Davarzani (UM), Biomarker discovery in heart failure
	16	Jaebok Kim (UT), Automatic recognition of engagement and emotion in a group of children
	17	Jianpeng Zhang (TU/e), On Graph Sample Clustering
	18	Henriette Nakad (UL), De Notaris en Private Rechtspraak
	19	Minh Duc Pham (VUA), Emergent relational schemas for RDF
	20	Manxia Liu (RUN), Time and Bayesian Networks
	21	Aad Slootmaker (OU), EMERGO: a generic platform for authoring and playing scenario-based serious games
	22	Eric Fernandes de Mello Araújo (VUA), Contagious: Modeling the Spread of Behaviours, Perceptions and Emotions in Social Networks
	23	Kim Schouten (EUR), Semantics-driven Aspect-Based Sentiment Analysis
	24	Jered Vroon (UT), Responsive Social Positioning Behaviour for Semi-Autonomous Telepresence Robots
	25	Riste Gligorov (VUA), Serious Games in Audio-Visual Collections
	26	Roelof Anne Jelle de Vries (UT),Theory-Based and Tailor-Made: Motivational Messages for Behavior Change Technology
	27	Maikel Leemans (TU/e), Hierarchical Process Mining for Scalable Software Analysis
	28	Christian Willemse (UT), Social Touch Technologies: How they feel and how they make you feel
	29	Yu Gu (TiU), Emotion Recognition from Mandarin Speech
	30	Wouter Beek (VUA), The ”K” in ”semantic web” stands for ”knowledge”: scaling semantics to the web

2019 01 Rob van Eijk (UL),Web privacy measurement in real-time bidding systems. A graph-based approach to RTB system classification
02 Emmanuelle Beauxis Aussalet (CWI, UU), Statistics and Visualizations for Assessing Class Size Uncertainty
03 Eduardo Gonzalez Lopez de Murillas (TU/e), Process Mining on Databases: Extracting Event Data from Real Life Data Sources
04 Ridho Rahmadi (RUN), Finding stable causal structures from clinical data
05 Sebastiaan van Zelst (TU/e), Process Mining with Streaming Data
06 Chris Dijkshoorn (VUA), Nichesourcing for Improving Access to Linked Cultural Heritage Datasets
07 Soude Fazeli (TUD), Recommender Systems in Social Learning Platforms
08 Frits de Nijs (TUD), Resource-constrained Multi-agent Markov Decision Processes
09 Fahimeh Alizadeh Moghaddam (UvA), Self-adaptation for energy efficiency in software systems
10 Qing Chuan Ye (EUR), Multi-objective Optimization Methods for Allocation and Prediction
11 Yue Zhao (TUD), Learning Analytics Technology to Understand Learner Behavioral Engagement in MOOCs
12 Jacqueline Heinerman (VUA), Better Together
13 Guanliang Chen (TUD), MOOC Analytics: Learner Modeling and Content Generation
14 Daniel Davis (TUD), Large-Scale Learning Analytics: Modeling Learner Behavior & Improving Learning Outcomes in Massive Open Online Courses
15 Erwin Walraven (TUD), Planning under Uncertainty in Constrained and Partially Observable Environments
16 Guangming Li (TU/e), Process Mining based on Object-Centric Behavioral Constraint (OCBC) Models
17 Ali Hurriyetoglu (RUN),Extracting actionable information from microtexts
18 Gerard Wagenaar (UU), Artefacts in Agile Team Communication
19 Vincent Koeman (TUD), Tools for Developing Cognitive Agents
20 Chide Groenouwe (UU), Fostering technically augmented human collective intelligence
21 Cong Liu (TU/e), Software Data Analytics: Architectural Model Discovery and Design Pattern Detection
22 Martin van den Berg (VUA),Improving IT Decisions with Enterprise Architecture
23 Qin Liu (TUD), Intelligent Control Systems: Learning, Interpreting, Verification
24 Anca Dumitrache (VUA), Truth in Disagreement - Crowdsourcing Labeled Data for Natural Language Processing
25 Emiel van Miltenburg (VUA), Pragmatic factors in (automatic) image description
26 Prince Singh (UT), An Integration Platform for Synchromodal Transport
27 Alessandra Antonaci (OU), The Gamification Design Process applied to (Massive) Open Online Courses
28 Esther Kuindersma (UL), Cleared for take-off: Game-based learning to prepare airline pilots for critical situations
29 Daniel Formolo (VUA), Using virtual agents for simulation and training of social skills in safety-critical circumstances
30 Vahid Yazdanpanah (UT), Multiagent Industrial Symbiosis Systems
31 Milan Jelisavcic (VUA), Alive and Kicking: Baby Steps in Robotics
32 Chiara Sironi (UM), Monte-Carlo Tree Search for Artificial General Intelligence in Games
33 Anil Yaman (TU/e), Evolution of Biologically Inspired Learning in Artificial Neural Networks

34 Negar Ahmadi (TU/e), EEG Microstate and Functional Brain Network Features for Classification of Epilepsy and PNES
35 Lisa Facey-Shaw (OU), Gamification with digital badges in learning programming
36 Kevin Ackermans (OU), Designing Video-Enhanced Rubrics to Master Complex Skills
37 Jian Fang (TUD), Database Acceleration on FPGAs
38 Akos Kadar (OU), Learning visually grounded and multilingual representations

2020 01 Armon Toubman (UL), Calculated Moves: Generating Air Combat Behaviour
02 Marcos de Paula Bueno (UL), Unraveling Temporal Processes using Probabilistic Graphical Models
03 Mostafa Deghani (UvA), Learning with Imperfect Supervision for Language Understanding
04 Maarten van Gompel (RUN), Context as Linguistic Bridges
05 Yulong Pei (TU/e), On local and global structure mining
06 Preethu Rose Anish (UT), Stimulation Architectural Thinking during Requirements Elicitation - An Approach and Tool Support
07 Wim van der Vegt (OU), Towards a software architecture for reusable game components
08 Ali Mirsoleimani (UL),Structured Parallel Programming for Monte Carlo Tree Search
09 Myriam Traub (UU), Measuring Tool Bias and Improving Data Quality for Digital Humanities Research
10 Alifah Syamsiyah (TU/e), In-database Preprocessing for Process Mining
11 Sepideh Mesbah (TUD), Semantic-Enhanced Training Data AugmentationMethods for Long-Tail Entity Recognition Models
12 Ward van Breda (VUA), Predictive Modeling in E-Mental Health: Exploring Applicability in Personalised Depression Treatment
13 Marco Virgolin (CWI), Design and Application of Gene-pool Optimal Mixing Evolutionary Algorithms for Genetic Programming
14 Mark Raasveldt (CWI/UL), Integrating Analytics with Relational Databases
15 Konstantinos Georgiadis (OU), Smart CAT: Machine Learning for Configurable Assessments in Serious Games
16 Ilona Wilmont (RUN), Cognitive Aspects of Conceptual Modelling
17 Daniele Di Mitri (OU), The Multimodal Tutor: Adaptive Feedback from Multimodal Experiences
18 Georgios Methenitis (TUD), Agent Interactions & Mechanisms in Markets with Uncertainties: Electricity Markets in Renewable Energy Systems
19 Guido van Capelleveen (UT), Industrial Symbiosis Recommender Systems
20 Albert Hankel (VUA), Embedding Green ICT Maturity in Organisations
21 Karine da Silva Miras de Araujo (VUA), Where is the robot?: Life as it could be
22 Maryam Masoud Khamis (RUN), Understanding complex systems implementation through a modeling approach: the case of e-government in Zanzibar
23 Rianne Conijn (UT), The Keys to Writing: A writing analytics approach to studying writing processes using keystroke logging
24 Lenin da Nóbrega Medeiros (VUA/RUN), How are you feeling, human? Towards emotionally supportive chatbots
25 Xin Du (TU/e), The Uncertainty in Exceptional Model Mining
26 Krzysztof Leszek Sadowski (UU), GAMBIT: Genetic Algorithm for Model-Based mixed-Integer opTimization
27 Ekaterina Muravyeva (TUD), Personal data and informed consent in an educational context
28 Bibeg Limbu (TUD), Multimodal interaction for deliberate practice: Training complex skills with augmented reality

29 Ioan Gabriel Bucur (RUN), Being Bayesian about Causal Inference
30 Bob Zadok Blok (UL), Creatief, Creatiever, Creatiefst
31 Gongjin Lan (VUA), Learning better – From Baby to Better
32 Jason Rhuggenaath (TU/e), Revenue management in online markets: pricing and online advertising
33 Rick Gilsing (TU/e), Supporting service-dominant business model evaluation in the context of business model innovation
34 Anna Bon (UM), Intervention or Collaboration? Redesigning Information and Communication Technologies for Development
35 Siamak Farshidi (UU), Multi-Criteria Decision-Making in Software Production

2021 01 Francisco Xavier Dos Santos Fonseca (TUD),Location-based Games for Social Interaction in Public Space
02 Rijk Mercuur (TUD), Simulating Human Routines: Integrating Social Practice Theory in Agent-Based Models
03 Seyyed Hadi Hashemi (UvA), Modeling Users Interacting with Smart Devices
04 Ioana Jivet (OU), The Dashboard That Loved Me: Designing adaptive learning analytics for self-regulated learning
05 Davide Dell'Anna (UU), Data-Driven Supervision of Autonomous Systems
06 Daniel Davison (UT), "Hey robot, what do you think?" How children learn with a social robot
07 Armel Lefebvre (UU), Research data management for open science
08 Nardie Fanchamps (OU), The Influence of Sense-Reason-Act Programming on Computational Thinking
09 Cristina Zaga (UT), The Design of Robothings. Non-Anthropomorphic and Non-Verbal Robots to Promote Children's Collaboration Through Play
10 Quinten Meertens (UvA), Misclassification Bias in Statistical Learning
11 Anne van Rossum (UL), Nonparametric Bayesian Methods in Robotic Vision
12 Lei Pi (UL), External Knowledge Absorption in Chinese SMEs
13 Bob R. Schadenberg (UT), Robots for Autistic Children: Understanding and Facilitating Predictability for Engagement in Learning
14 Negin Samaeemofrad (UL), Business Incubators: The Impact of Their Support
15 Onat Ege Adali (TU/e), Transformation of Value Propositions into Resource Re-Configurations through the Business Services Paradigm
16 Esam A. H. Ghaleb (UM), Bimodal emotion recognition from audio-visual cues
17 Dario Dotti (UM), Human Behavior Understanding from motion and bodily cues using deep neural networks
18 Remi Wieten (UU), Bridging the Gap Between Informal Sense-Making Tools and Formal Systems - Facilitating the Construction of Bayesian Networks and Argumentation Frameworks
19 Roberto Verdecchia (VUA), Architectural Technical Debt: Identification and Management
20 Masoud Mansoury (TU/e), Understanding and Mitigating Multi-Sided Exposure Bias in Recommender Systems
21 Pedro Thiago Timbó Holanda (CWI), Progressive Indexes
22 Sihang Qiu (TUD), Conversational Crowdsourcing
23 Hugo Manuel Proença (UL), Robust rules for prediction and description
24 Kaijie Zhu (TU/e), On Efficient Temporal Subgraph Query Processing
25 Eoin Martino Grua (VUA), The Future of E-Health is Mobile: Combining AI and Self-Adaptation to Create Adaptive E-Health Mobile Applications

26 Benno Kruit (CWI/VUA), Reading the Grid: Extending Knowledge Bases from Human-readable Tables
27 Jelte van Waterschoot (UT), Personalized and Personal Conversations: Designing Agents Who Want to Connect With You
28 Christoph Selig (UL), Understanding the Heterogeneity of Corporate Entrepreneurship Programs

2022 01 Judith van Stegeren (UT), Flavor text generation for role-playing video games
02 Paulo da Costa (TU/e), Data-driven Prognostics and Logistics Optimisation: A Deep Learning Journey
03 Ali el Hassouni (VUA), A Model A Day Keeps The Doctor Away: Reinforcement Learning For Personalized Healthcare
04 Ünal Aksu (UU), A Cross-Organizational Process Mining Framework
05 Shiwei Liu (TU/e), Sparse Neural Network Training with In-Time Over-Parameterization
06 Reza Refaei Afshar (TU/e), Machine Learning for Ad Publishers in Real Time Bidding
07 Sambit Praharaj (OU), Measuring the Unmeasurable? Towards Automatic Co-located Collaboration Analytics
08 Maikel L. van Eck (TU/e), Process Mining for Smart Product Design
09 Oana Andreea Inel (VUA), Understanding Events: A Diversity-driven Human-Machine Approach
10 Felipe Moraes Gomes (TUD), Examining the Effectiveness of Collaborative Search Engines
11 Mirjam de Haas (UT), Staying engaged in child-robot interaction, a quantitative approach to studying preschoolers' engagement with robots and tasks during second-language tutoring
12 Guanyi Chen (UU), Computational Generation of Chinese Noun Phrases
13 Xander Wilcke (VUA), Machine Learning on Multimodal Knowledge Graphs: Opportunities, Challenges, and Methods for Learning on Real-World Heterogeneous and Spatially-Oriented Knowledge
14 Michiel Overeem (UU), Evolution of Low-Code Platforms
15 Jelmer Jan Koorn (UU), Work in Process: Unearthing Meaning using Process Mining
16 Pieter Gijsbers (TU/e), Systems for AutoML Research
17 Laura van der Lubbe (VUA), Empowering vulnerable people with serious games and gamification
18 Paris Mavromoustakos Blom (TiU), Player Affect Modelling and Video Game Personalisation
19 Bilge Yigit Ozkan (UU), Cybersecurity Maturity Assessment and Standardisation
20 Fakhra Jabeen (VUA), Dark Side of the Digital Media - Computational Analysis of Negative Human Behaviors on Social Media
21 Seethu Mariyam Christopher (UM), Intelligent Toys for Physical and Cognitive Assessments
22 Alexandra Sierra Rativa (TiU), Virtual Character Design and its potential to foster Empathy, Immersion, and Collaboration Skills in Video Games and Virtual Reality Simulations
23 Ilir Kola (TUD), Enabling Social Situation Awareness in Support Agents
24 Samaneh Heidari (UU), Agents with Social Norms and Values - A framework for agent based social simulations with social norms and personal values
25 Anna L.D. Latour (UL), Optimal decision-making under constraints and uncertainty
26 Anne Dirkson (UL), Knowledge Discovery from Patient Forums: Gaining novel medical insights from patient experiences
27 Christos Athanasiadis (UM), Emotion-aware cross-modal domain adaptation in video sequences

28 Onuralp Ulusoy (UU), Privacy in Collaborative Systems
29 Jan Kolkmeier (UT), From Head Transform to Mind Transplant: Social Interactions in Mixed Reality
30 Dean De Leo (CWI), Analysis of Dynamic Graphs on Sparse Arrays
31 Konstantinos Traganos (TU/e), Tackling Complexity in Smart Manufacturing with Advanced Manufacturing Process Management
32 Cezara Pastrav (UU), Social simulation for socio-ecological systems
33 Brinn Hekkelman (CWI/TUD), Fair Mechanisms for Smart Grid Congestion Management
34 Nimat Ullah (VUA), Mind Your Behaviour: Computational Modelling of Emotion & Desire Regulation for Behaviour Change
35 Mike E.U. Ligthart (VUA), Shaping the Child-Robot Relationship: Interaction Design Patterns for a Sustainable Interaction

---

2023 01 Bojan Simoski (VUA), Untangling the Puzzle of Digital Health Interventions
02 Mariana Rachel Dias da Silva (TiU), Grounded or in flight? What our bodies can tell us about the whereabouts of our thoughts
03 Shabnam Najafian (TUD), User Modeling for Privacy-preserving Explanations in Group Recommendations
04 Gineke Wiggers (UL), The Relevance of Impact: bibliometric-enhanced legal information retrieval
05 Anton Bouter (CWI), Optimal Mixing Evolutionary Algorithms for Large-Scale Real-Valued Optimization, Including Real-World Medical Applications
06 António Pereira Barata (UL), Reliable and Fair Machine Learning for Risk Assessment
07 Tianjin Huang (TU/e), The Roles of Adversarial Examples on Trustworthiness of Deep Learning
08 Lu Yin (TU/e), Knowledge Elicitation using Psychometric Learning
09 Xu Wang (VUA), Scientific Dataset Recommendation with Semantic Techniques
10 Dennis J.N.J. Soemers (UM), Learning State-Action Features for General Game Playing
11 Fawad Taj (VUA), Towards Motivating Machines: Computational Modeling of the Mechanism of Actions for Effective Digital Health Behavior Change Applications
12 Tessel Bogaard (VUA), Using Metadata to Understand Search Behavior in Digital Libraries
13 Injy Sarhan (UU), Open Information Extraction for Knowledge Representation
14 Selma Čaušević (TUD), Energy resilience through self-organization
15 Alvaro Henrique Chaim Correia (TU/e), Insights on Learning Tractable Probabilistic Graphical Models
16 Peter Blomsma (TiU), Building Embodied Conversational Agents: Observations on human nonverbal behaviour as a resource for the development of artificial characters
17 Meike Nauta (UT), Explainable AI and Interpretable Computer Vision – From Oversight to Insight
18 Gustavo Penha (TUD), Designing and Diagnosing Models for Conversational Search and Recommendation
19 George Aalbers (TiU), Digital Traces of the Mind: Using Smartphones to Capture Signals of Well-Being in Individuals
20 Arkadiy Dushatskiy (TUD), Expensive Optimization with Model-Based Evolutionary Algorithms applied to Medical Image Segmentation using Deep Learning
21 Gerrit Jan de Bruin (UL), Network Analysis Methods for Smart Inspection in the Transport Domain
22 Alireza Shojaifar (UU), Volitional Cybersecurity
23 Theo Theunissen (UU), Documentation in Continuous Software Development

24 Agathe Balayn (TUD), Practices Towards Hazardous Failure Diagnosis in Machine Learning
25 Jurian Baas (UU), Entity Resolution on Historical Knowledge Graphs
26 Loek Tonnaer (TU/e), Linearly Symmetry-Based Disentangled Representations and their Out-of-Distribution Behaviour
27 Ghada Sokar (TU/e), Learning Continually Under Changing Data Distributions
28 Floris den Hengst (VUA), Learning to Behave: Reinforcement Learning in Human Contexts
29 Tim Draws (TUD), Understanding Viewpoint Biases in Web Search Results

2024 01 Daphne Miedema (TU/e), On Learning SQL: Disentangling concepts in data systems education
02 Emile van Krieken (VUA), Optimisation in Neurosymbolic Learning Systems
03 Feri Wijayanto (RUN), Automated Model Selection for Rasch and Mediation Analysis
04 Mike Huisman (UL), Understanding Deep Meta-Learning
05 Yiyong Gou (UM), Aerial Robotic Operations: Multi-environment Cooperative Inspection & Construction Crack Autonomous Repair
06 Azqa Nadeem (TUD), Understanding Adversary Behavior via XAI: Leveraging Sequence Clustering to Extract Threat Intelligence
07 Parisa Shayan (TiU), Modeling User Behavior in Learning Management Systems
08 Xin Zhou (UvA), From Empowering to Motivating: Enhancing Policy Enforcement through Process Design and Incentive Implementation
09 Giso Dal (UT), Probabilistic Inference Using Partitioned Bayesian Networks
10 Cristina-Iulia Bucur (VUA), Linkflows: Towards Genuine Semantic Publishing in Science
11 withdrawn
12 Peide Zhu (TUD), Towards Robust Automatic Question Generation For Learning
13 Enrico Liscio (TUD), Context-Specific Value Inference via Hybrid Intelligence
14 Larissa Capobianco Shimomura (TU/e), On Graph Generating Dependencies and their Applications in Data Profiling
15 Ting Liu (VUA), A Gut Feeling: Biomedical Knowledge Graphs for Interrelating the Gut Microbiome and Mental Health
16 Arthur Barbosa Câmara (TUD), Designing Search-as-Learning Systems
17 Razieh Alidoosti (VUA), Ethics-aware Software Architecture Design
18 Laurens Stoop (UU), Data Driven Understanding of Energy-Meteorological Variability and its Impact on Energy System Operations
19 Azadeh Mozafari Mehr (TU/e), Multi-perspective Conformance Checking: Identifying and Understanding Patterns of Anomalous Behavior
20 Ritsart Anne Plantenga (UL), Omgang met Regels
21 Federica Vinella (UU), Crowdsourcing User-Centered Teams
22 Zeynep Ozturk Yurt (TU/e), Beyond Routine: Extending BPM for Knowledge-Intensive Processes with Controllable Dynamic Contexts
23 Jie Luo (VUA), Lamarck's Revenge: Inheritance of Learned Traits Improves Robot Evolution
24 Nirmal Roy (TUD), Exploring the effects of interactive interfaces on user search behaviour
25 Alisa Rieger (TUD), Striving for Responsible Opinion Formation in Web Search on Debated Topics
26 Tim Gubner (CWI), Adaptively Generating Heterogeneous Execution Strategies using the VOILA Framework
27 Lincen Yang (UL), Information-theoretic Partition-based Models for Interpretable Machine Learning

28 Leon Helwerda (UL), Grip on Software: Understanding development progress of Scrum sprints and backlogs
29 David Wilson Romero Guzman (VUA), The Good, the Efficient and the Inductive Biases: Exploring Efficiency in Deep Learning Through the Use of Inductive Biases
30 Vijanti Ramautar (UU), Model-Driven Sustainability Accounting
31 Ziyu Li (TUD), On the Utility of Metadata to Optimize Machine Learning Workflows
32 Vinicius Stein Dani (UU), The Alpha and Omega of Process Mining
33 Siddharth Mehrotra (TUD), Designing for Appropriate Trust in Human-AI interaction
34 Robert Deckers (VUA), From Smallest Software Particle to System Specification - MuD-ForM: Multi-Domain Formalization Method
35 Sicui Zhang (TU/e), Methods of Detecting Clinical Deviations with Process Mining: a fuzzy set approach
36 Thomas Mulder (TU/e), Optimization of Recursive Queries on Graphs
37 James Graham Nevin (UvA), The Ramifications of Data Handling for Computational Models
38 Christos Koutras (TUD), Tabular Schema Matching for Modern Settings
39 Paola Lara Machado (TU/e), The Nexus between Business Models and Operating Models: From Conceptual Understanding to Actionable Guidance
40 Montijn van de Ven (TU/e), Guiding the Definition of Key Performance Indicators for Business Models
41 Georgios Siachamis (TUD), Adaptivity for Streaming Dataflow Engines
42 Emmeke Veltmeijer (VUA), Small Groups, Big Insights: Understanding the Crowd through Expressive Subgroup Analysis
43 Cedric Waterschoot (KNAW Meertens Instituut), The Constructive Conundrum: Computational Approaches to Facilitate Constructive Commenting on Online News Platforms
44 Marcel Schmitz (OU), Towards learning analytics-supported learning design
45 Sara Salimzadeh (TUD), Living in the Age of AI: Understanding Contextual Factors that Shape Human-AI Decision-Making
46 Georgios Stathis (Leiden University), Preventing Disputes: Preventive Logic, Law & Technology
47 Daniel Daza (VUA), Exploiting Subgraphs and Attributes for Representation Learning on Knowledge Graphs
48 Ioannis Petros Samiotis (TUD), Crowd-Assisted Annotation of Classical Music Compositions

---

2025 01 Max van Haastrecht (UL), Transdisciplinary Perspectives on Validity: Bridging the Gap Between Design and Implementation for Technology-Enhanced Learning Systems
02 Jurgen van den Hoogen (JADS), Time Series Analysis Using Convolutional Neural Networks
03 Andra-Denis Ionescu (TUD), Feature Discovery for Data-Centric AI
04 Rianne Schouten (TU/e), Exceptional Model Mining for Hierarchical Data
05 Nele Albers (TUD), Psychology-Informed Reinforcement Learning for Situated Virtual Coaching in Smoking Cessation
06 Daniël Vos (TUD), Decision Tree Learning: Algorithms for Robust Prediction and Policy Optimization
07 Ricky Maulana Fajri (TU/e), Towards Safer Active Learning: Dealing with Unwanted Biases, Graph-Structured Data, Adversary, and Data Imbalance
08 Stefan Bloemheuvel (TiU), Spatio-Temporal Analysis Through Graphs: Predictive Modeling and Graph Construction
09 Fadime Kaya (VUA), Decentralized Governance Design - A Model-Based Approach

10 Zhao Yang (UL), Enhancing Autonomy and Efficiency in Goal-Conditioned Reinforcement Learning
11 Shahin Sharifi Noorian (TUD), From Recognition to Understanding: Enriching Visual Models Through Multi-Modal Semantic Integration
12 Lijun Lyu (TUD), Interpretability in Neural Information Retrieval
13 Fuda van Diggelen (VUA), Robots Need Some Education: on the complexity of learning in evolutionary robotics
14 Gennaro Gala (TU/e), Probabilistic Generative Modeling with Latent Variable Hierarchies
15 Michiel van der Meer (UL), Opinion Diversity through Hybrid Intelligence

Data Management

The research in the thesis has been carried out under the ICIS research data management policy. The policy and protocol were accessed on 3 July 2023 from https://www.ru.nl/icis/research-data-management/policy-protocol/.

Datasets and software used in this thesis are publicly available and accessible. The dataset for Part I (Cumulative Citation Recommendation) is well documented as part of NIST's standards on evaluations and described in the following two papers:

- J. R. Frank, M. Kleiman-Weiner, D. A. Roberts, F. Niu, C. Zhang, C. Ré, and I. Soboroff, Building an Entity-centric Stream Filtering Test Collection for TREC 2012, Tech. Rep. (Massachusetts Institute of Technology Cambridge, 2012).
- J. R. Frank, S. J. Bauer, M. Kleiman-Weiner, D. A. Roberts, N. Tripuraneni, C. Zhang, C. Re, E. Voorhees, and I. Soboroff, Evaluating Stream Filtering for Entity Profile Updates for TREC 2013 (KBA Track Overview), Tech. Rep. (Massachusetts Inst of Tech Cambridge, 2013).

The software tools we used are also publicly available and documented, and they are well-described in the respective chapters. The dataset we used to represent entities can be found at

- Valentin I. Spitkovsky and Angel X. Chang. A Cross-Lingual Dictionary for English Wikipedia Concepts In Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC 2012), pages 3168–3175, Istanbul, Turkey. European Language Resources Association (ELRA).

The datasets and platforms for Part II and Part III (New Recommendation and Measuring Recommender System Personalization) were from CLEF NewsREEL and as such are described in the following documents.

- Benjamin Kille, Andreas Lommatzsch, Roberto Turrin, Andr´as Ser´eny, Martha Larson, Torben Brodt, Jonas Seiler, and Frank Hopfgartner. Overview of CLEF NewsREEL 2015: News Recommendation Evaluation Lab. In Working Notes of CLEF 2015 - Conference and Labs of the Evaluation forum, Toulouse, France, September 8-11, 2015.

- Benjamin Kille, Andreas Lommatzsch, Frank Hopfgartner, Martha Larson, Jonas Seiler, Davide Malagoli, Andras Sereny, and Torben Brodt. CLEF NewsREEL 2016: Comparing Multi-dimensional Offline and Online Evaluation of News Recommender Systems. In Working Notes of CLEF 2016 - Conference and Labs of the Evaluation forum, Évora, Portugal, pages 593–605, September 2016.
- Torben Brodt and Frank Hopfgartner. Shedding Light on a Living Lab: the CLEF NEWSREEL open recommendation platform. In Proceedings of the 5th Information Interaction in Context Symposium (2014). Association for Computing Machinery, New York, NY, USA.

Links to our versions of algorithms used in Part II are below:

- Recency Algorithm: https://github.com/gebre/recency
- RecencyRandom: https://github.com/gebre/recencyRandom
- Recency with Geographic info: https://github.com/gebre/georec

Summary

This thesis sheds light on the different components of the recommendation pipeline, under three themes, which are divided in 10 chapters. The first theme is Cumulative Citation Recommendation. Under this theme, we have conducted research on the task of Cumulative Citation Recommendation (CCR), which is the automation and maintenance of knowledge bases such as Wikipedia. Given a set of Knowledge Base entities, CCR is the task of filtering and ranking documents according to their citation worthiness to the entities. We specifically focused on the filtering stage of the recommendation process and the interplay between feature sets and machine learning algorithms.

There are four chapters under the first theme: Chapters 3 to 6. Chapter 3 presents experiments with string-matching and machine learning approaches to the task of CCR. Chapter 4 investigates the interplay between the choice of feature sets and their impact on the performance of machine learning algorithms. Chapter 5 investigates the impact of the initial task of filtering in the CCR overall performance, and what makes some documents unfilterable. Chapter 6 reviews new advances in the area of the theme and the specific chapters. Under this theme, we show that simple string-matching approaches can have advantages over complex machine learning approaches for the task of CCR, that comparisons of machine learning algorithms should take into account the sets of features used, and that the filtering stage of a CCR task can impact recommender systems performance in different ways.

The second theme is News Recommendation. In this theme, we investigate news recommendation with a particular focus on evaluation. We study the role of geography in news consumption to understand the geographical focus of news items and the geographical location of readers followed by the incorporation of geographic information into online deployments of algorithms. We also attempt to quantify random fluctuations in the performance difference of a live recommender system. After that, we focus on news evaluation, investigating it from several angles. We conducted A/A tests (running two instances of the same algorithm), offline evaluations, online evaluations, and comparisons of algorithm performances across years.

There are three chapters under the theme of News Recommendation. Chapter 7 investigates the role of geographic information in news consumption, and examines in a real-world setting, the performance patterns of news recommender systems, one of which incorporates geographic information into its algorithm. Chapter 8 examines the challenges, validity, and consistency of news recommender systems evaluations from multiple perspectives, involving A/A tests, offline evaluations, online evaluations, and comparisons of algorithm performances across years. Chapter 9 reviews advances in News Recommendation with a focus on developments that have relevance to the approaches and findings presented in chapters 7 and 8.

In the above theme, we show that user and item geography play a role in the consumption of news, that there are significant differences and discrepancies in offline and online evaluation of recommender systems algorithms, and that random effects on online performances

can result in statistically significant performance differences.

The third and final theme is Measuring Personalization and consists of Chapter 10. We view personalization as introducing or imposing differentiation between users in terms of the items recommended to them. In the differentiation, some items will be shared between users, and some will not. We then propose and apply a user-centric metric of personalization that, by using the recommendation lists and the resulting user reaction lists that result from users choosing to click or react on, measures the degree of users' tendency to agree to the differentiation introduced or imposed between them by the recommender system, to converge (by, for example, clicking more on shared items), or to diverge from the differentiation (by, for example, clicking more on the items that are not in shared recommendation).

Samenvatting

Dit proefschrift belicht verschillende componenten van de recommendation pipeline via drie thema's, onderverdeeld in 10 hoofdstukken. Het eerste thema, Cumulative Citation Recommendation,bestudeert het cumulatief aanbevelen van citaties. Bij dit thema hebben we onderzoek gedaan naar de Cumulatieve Citation Recommendation (CCR) taak, de automatisering en het onderhoud van knowledge bases zoals Wikipedia. Gegeven een set Knowledge Base entiteiten, is CCR de taak waarbij documenten gefilterd en gerangschikt worden op basis van hun citatiewaardigheid voor de entiteiten. Het onderzoek richt zich specifiek op de filterfase van het aanbevelingsproces en op de wisselwerking tussen featuresets en machine learning-algoritmen.

Het eerste thema is onderverdeeld in vier hoofdstukken: hoofdstuk 3 tot en met 6. Hoofdstuk 3 presenteert experimenten met string-matching en machine learning-benaderingen voor de CCRtaak. Hoofdstuk 4 kijkt naar de wisselwerking tussen de keuze van featuresets en hun impact op de prestaties van machine learning-algoritmen. Hoofdstuk 5 onderzoekt de impact van de initiële filtertaak op de algehele CCRprestaties en de aspecten die het onmogelijk maken om sommige documenten te filteren. Hoofdstuk 6 bespreekt nieuwe ontwikkelingen op het gebied van het thema en de specifieke hoofdstukken. Onder dit thema laten we zien dat voor de CCRtaak eenvoudige string-matching-benaderingen voordelen kunnen hebben ten opzichte van complexe machine learning-benaderingen. We laten ook zien dat er bij vergelijkingen van machine learning-algoritmen rekening moet worden gehouden met de sets van gebruikte features en dat de filterfase van een CCRtaak de prestaties van aanbevelingssystemen op verschillende manieren kan beïnvloeden.

Het tweede thema is News Recommendation. In dit thema doen we onderzoek naar nieuwsaanbevelingen, met een specifieke focus op evaluatie. We onderzochten de rol van geografie in nieuwsconsumptie om de geografische focus van nieuwsitems en de locatie van lezers te begrijpen, gevolgd door de integratie van geografische informatie in online implementaties van algoritmen. We probeerden ook om willekeurige fluctuaties in het prestatieverschil van een live aanbevelingssysteem te kwantificeren. Daarna richten we ons op nieuwsevaluatie en onderzochten dit vanuit verschillende invalshoeken. We voerden A/A-tests uit (twee instanties van hetzelfde algoritme), offline-evaluaties, online-evaluaties en vergeleken prestaties van algoritmes door de jaren heen.

Er vallen drie hoofdstukken onder het thema News Recommendation. Hoofdstuk 7 doet onderzoek naar de rol van geografische informatie in nieuwsconsumptie. We onderzochten in een 'real-world' setting de prestatiepatronen van nieuwsaanbevelingssystemen; bij één ervan was geografische informatie in het algoritme verwerkt. Hoofdstuk 8 focust zich op de uitdagingen, validiteit en consistentie van evaluaties van nieuwsaanbevelingssystemen vanuit meerdere perspectieven, met inbegrip van A/A-tests, offline evaluaties, online evaluaties en vergelijkingen van prestaties van algoritmen door de jaren heen. Hoofdstuk 9 bespreekt ontwikkelingen in nieuwsaanbevelingen met een specifieke focus op ontwikkelingen die relevant zijn voor de benaderingen en bevindingen die in de hoofdstukken 7 en 8

worden gepresenteerd.

In het bovenstaande thema laten we zien dat gebruikers- en item-geografie een rol spelen in de consumptie van nieuws, dat er significante verschillen resultatenzijn tussen offline en online evaluatie van algoritmen van aanbevelingssystemen, en dat willekeurige effecten op online prestaties kunnen resulteren in statistisch significante prestatieverschillen.

Het derde en laatste thema is Measuring Personalization: het meten aan personalisatie. Dit thema is beschreven in hoofdstuk 10. We zien personalisatie als differentiatie tussen gebruikers in termen van de items die aan hen worden aanbevolen. In de differentiatie worden sommige items gedeeld tussen gebruikers en andere niet. Daarna we stellen aan gebruikersgerichte metriek voor personalisatie voor en passen deze toe. Daarbij gebruiken we de aanbevelingslijsten en de lijsten van gebruikersreacties die het resultaat zijn van de keuze van gebruikers om te klikken of te reageren. Hiermee meet de metriek in welke mate gebruikers geneigd zijn om de differentiatie die het aanbevelingssysteem tussen hen introduceert of oplegt te accepteren, te convergeren (door bijvoorbeeld meer te klikken op gedeelde items), of te divergeren van de differentiatie (door bijvoorbeeld meer op de items te klikken die geen gedeelde aanbeveling zijn).

Acknowledgments

My PhD journey has been an arduous ordeal, shaped by a range of enormous challenges involving complicated workplace conditions, prolonged limbo, disorientation, and most profoundly by the genocidal war on Tigray and Tigrayans, which erupted during the final and most crucial stretch of this work. Despite everything, I was able to persevere and complete this journey thanks to the support of many people.

I thank my supervisor, Arjen P. de Vries, for his support and guidance and for seeing this project through with me, despite the challenges along the way. I also want to acknowledge my former colleagues at CWI for the time and experiences we shared, and Henriette Cramer for the enriching internship opportunity at Yahoo! Labs in Sunnyvale, USA.

I am especially grateful to Anne and my son Gideon (11), who stood by me through the most difficult times—sharing in my struggle, and surrounding me with love, patience, and encouragement. Their support helped carry me through, even during the painful two-year period when I was completely cut off from my family due to the communications blackout in Tigray and the overall devastation of the war.

My parents and siblings have played an important role in shaping the person I am. They've always believed in my educational pursuits, trusting that every step I took was a step forward. My brothers and sisters—often asking, "When are you finishing the PhD?"—helped keep me focused and grounded in the importance of completing this work. I thank them for their belief in me.

I am deeply thankful to my friend Teklai Gebremichael, who encouraged me to finish the PhD, and who stood as a steadfast companion in our shared resistance to the war on Tigray. Together, we wrote, spoke, and bore witness through platforms like Tghat, amplifying the voices of our people and enduring the pain of war, blockade, and loss.

I also extend my gratitude to my Tghat colleagues, and to all those who fought courageously—both on the battlefields in Tigray and in the global diaspora advocacy efforts. Their courage and determination gave me hope and strength when I needed it.

This dissertation is dedicated to the memory of those we lost, to those who endured, and to all who continue to fight for truth, justice, and the dignity of our people.

Curriculum Vitae

Gebrekirstos Gebreselassie Gebremeskel was born in Tigray, Ethiopia. He spent his early years as a herder, gold panner, and a helping hand on the family farm, taking on various roles before beginning his formal education at Megab Elementary School in his later teenage years. He attended Kellamino Special High School for his secondary education.

In 2007, Gebrekirstos earned a BSc in Computer Science from Mekelle University, Tigray, Ethiopia. He briefly worked there as a graduate assistant before receiving the prestigious Erasmus Mundus scholarship in Language & Communication Technologies (EM LCT). Through this program, he completed an MSc in Human Language Science and Technology at the University of Malta, and an MA in Linguistics (Research Master) at the University of Groningen in the Netherlands, graduating in 2011.

From 2012 to 2017, he was a junior researcher in the Information Access Group at CWI in Amsterdam. During this time, he also completed an internship at Yahoo! Labs in the USA (July–November 2015). After his time at CWI, he worked at Geophy in Delft as an NLP Extraction Specialist. In 2017, he enrolled as an external PhD student at Radboud University Nijmegen.

Gebrekirstos is also the creator of mermru.com, a platform where he shares his linguistic and computational work on Tigrinya, Ge'ez, and Amharic. In response to the outbreak of the war on Tigray in 2020, he founded Tghat.com to document the war and advocate for peace, justice, and accountability. He continues to serve as Executive Director and Chief Editor of Tghat, while also working as a freelance data scientist and journalist.

Radboud Universiteit