Master Thesis

Designing Human-centered News Recommender Systems

Author: Dominic Istha (2600140)

1st supervisor:	Vincent François-Lavet	
2nd reader:	Nicolas Michael Mattis	
1st daily supervisor:	Niya Stoimenova	(DEUS)
2nd daily supervisor:	João Reis	(DEUS)

A thesis submitted in fulfillment of the requirements for the Master of Science degree in Business Analytics at the Vrije Universiteit Amsterdam.

August 18, 2023

Abstract

Recommender system research has been overly fixated on accuracy, which is severly limited and misguided [1; 2; 3; 4]. Therefore, this work advocates for an alternative approach to deep learning research for news recommender systems that considers the user-experience and user-satisfaction as fundamental objectives, instead of solely being guided by accuracy. This work contends that accuracy fails to account for the full meaning of user-satisfaction. Therefore, this work explores users' objectives and perception of news recommendations and news recommender systems to develop a better understanding of what users want. Beyond-accuracy evaluation methods are presented to promote a comprehensive framework for evaluating news recommender systems, which is more in line with user-satisfaction than accuracy alone. Furthermore, this thesis aims to take a broad perspective of news recommender systems. This is done on a systemic level, by providing a conceptual and theoretical review of two-tower deep learning models and multi-stage recommender systems. In addition, this thesis explores several broad perspectives on news recommender systems, by placing news recommender systems in their application context, organizational context, and societal context. In doing so, this work aims to facilitate insights and opportunities for interdisciplinary research on the crossover between deep learning and the aforementioned contexts. The theoretical work is complemented with a case study on the Microsoft News (MIND) dataset [5] to provide a practical starting point for researchers and practitioners (all code is made publicly available on GitHub[1]). Based on the experiments conducted, the following conclusions are drawn. First, current beyond-accuracy evaluation metrics are highly domain-dependent and the details greatly impact applicability, interpretability and reproducibility. Second, especially when latent representations are adopted to evaluate diversity, contradicting conclusions

[1] The code of this research project is available on GitHub at: https://github.com/DIstha/THESIS_PROJECT

may emerge depending on the method used to construct the latent representations. Thus, further research and user-studies are required to validate whether beyond-accuracy metrics coincide with users' perception of aspects such as diversity and novelty. Based on the theoretical and practical insights, this work aims to provide an alternative perspective on the way academics and practitioners think about researching and developing deep learning models for news recommender systems. Rather than adopting a traditional, vertical, modular research-approach, this work adopts a broad, lateral perspective, which may overcome certain limitations of traditional research.

Contents

List of Figures

List of Tables

Glossary

Abbreviations

AUC	Area Under the Curve
AUC	Area Under the ROC Curve
CF	Collaborative Filtering
DCG	Discounted Cumulative Gain
FN	False Negatives
FP	False Positives
IDCG	Ideal Discounted Cumulative Gain
ILS	Intra-List Similarity
KNN	k-Nearest Neighbours
LDA	Latent Dirichlet Allocation
LSH	Locality Sensitive Hashing
MAE	Mean Absolute Error
MIND	Microsoft News Dataset
MRR	Mean Reciprocal Rank
MSE	Mean Squared Error
nDCG	Normalized Discounted Cumulative Gain. nDCG5 or nDCG10 indicates that the metric is evaluated over the top 5 or top 10 items, respectively.
NLP	Natural Language Processing
NLP	Natural Language Processing
NRS	News Recommender System
RMSE	Root Mean Squared Error
ROC	Receiver Operating Characteristic curve
RS	Recommender System
SGD	Stochastic Gradient Descent

TF-IDF	Term Frequency - Inverse Document Frequency
TN	True Negatives
TP	True Positives

Definitions

Candidate Retrieval Candidate retrieval ("candidate generation" or simply "retrieval") is generally the first stage of a recommender system, where a light-weight system retrieves a subset of potentially relevant items for a particular user from a large collection of items [9].

List-wise ranking List-wise ranking approaches directly attempt to determine the optimal ordering of the entire list of candidate items.

Pair-wise ranking Pair-wise ranking approaches look at a pair of candidate items and attempt to optimize the ordering for that pair compared to the ground truth.

Point-wise scoring Point-wise scoring (or ranking) refers to machine learning models which determine a score or probability for the relevance of a single item at the time.

Ranking Ranking usually refers to the application of a point-wise scoring model (in that case, "scoring" may be used synonymous), pair-wise ranking model or list-wise ranking model. In some literature readers may find these ranking models also labeled as "recommender systems" or "recommender models", but to effectively delineate between the entire system and the ranking stage, I will refrain from doing so.

Re-ranking Re-ranking (or ordering) is done when the order of items is altered after an initial ordering has been established in the ranking stage, often with the goal of improving the utility of the entire list.

Recommender System (RS / RecSys) A recommender system or recommendation system (sometimes "ranking system" is used or synonyms such as "platform" or "engine" are used to replace "system") is a class of systems that provide suggestions for items which are relevant to a particular user [10].

Symbols

$\mathcal{T}$	Training set
$\mathcal{V}$	Set of features of news item
$\phi(u)$	User encoder
ϕ_u	Latent representation of user u
$\psi(v)$	News encoder
ψ_v	Latent representation of news item v
θ	Model parameters
B	Batch of training samples
C	Set of all candidate news items c

c	Candidate news item
D	Dataset
H_u	History of item interactions of user u
PM	Primitive Recommendation Model
Q	Set of queries / recommendation requests
q_i	Reward of news item i
R_u	List of recommendations for user u
$s(v, u)$	Two-tower deep neural recommendation model
T	Set of training samples
U	Set of all users u
u	User
V	Set of all news items v
v	News item

1

Introduction

In an era of information overload, news recommender systems are emerging as an essential tool for assisting users in navigating the vast landscape of news content. With the surge in online news sources and the diverse interests and preferences of users, personalized recommendations systems play a pivotal role in enhancing user satisfaction. By leveraging advanced algorithms and techniques to analyze users' preferences and historical behaviour, recommender systems are able to provide increasingly personalized experiences. News rcommender systems are able to automatically filter and curate news content, offering tailored recommendations that align with users their interests and preferences.

The field of news recommender systems has seen great advancements in recent years, by leveraging the increasing capabilities of deep learning models. The overwhelming majority of academic research on deep learning models for (news) recommender systems, thus far, is predominantly guided by accuracy. Accuracy as such is often measured using metrics like AUC, precision or recall (see Section 8.1). Using accuracy to guide research and development of deep learning models for news recommender systems is dependent on two fundamental assumptions, which are often taken for granted by academics in the field of computer science:

1. Accuracy metrics reflect which items users want to see.
2. Providing more accurate recommendations is better.

The degree to which these assumptions are entrenched in computer science is illustrated by the terminology used by academics in the field. Terms such as relevance, usefulness, utility or satisfaction are used synonymous for accuracy (e.g. [11]), without conclusive evidence that accurate recommendations are indeed perceived by users as relevant, useful, or satisfying.

Some researchers have argued that these assumptions may only be true to a certain degree and that taking these assumptions for granted (i.e using accuracy as a definitive measure for what users want or which items are relevant, useful or satisfying) is inherently limited [1; 2; 3; 4]. For example, previous research has shown that users may be inclined to choose less accurate items in an attempt to improve their overall experience using a recommender system [2; 3]. The works of Swearingen et al. [12] and Pu et al. [8] also demonstrate that accuracy may only be partially indicative of whether users want the recommended items or whether they perceive the system as useful. To illustrate, the work of Pu et al. demonstrates that to what extent users want a recommended item (as expressed by users through a questionnaire) may depend on several qualities of the recommendations, such as the accuracy, novelty and diversity of recommendations, as well as users' beliefs about the transparency, control, usefulness and ease of use of the system. To conclude, there are strong indications that accuracy only partially reflects to what degree users want an item and providing more accurate recommendations may not necessarily be the most efficient way to achieve *better* recommender systems.

This work builds upon the arguments that accuracy as a measure of what users want is inherently limited, in an attempt to take a step towards an alternative approach to research and development of deep learning models for news recommender systems. This work focuses on news recommender systems, because of the interests of the organizations affiliated with this research. However, many of the findings in this thesis may be applicable to recommender systems in other domains as well. This work is an exploration of how the field of computer science can research, develop and evaluate deep learning models for news recommendations without solely relying on accuracy as a definitive measure for what users want. This work will argue that advancing the field of news recommender systems with the use of deep learning models requires a design- and research-philosophy that puts humans at the centre of the process, i.e. a human-centered approach. *Human-centered design* may be defined as "an approach to interactive systems development that [...] enhances effectiveness and efficiency, improves human well-being, user satisfaction, accessibility and sustainability; and counteracts possible adverse effect of use on human health, safety and performance [13]." In other words, this work contends that designing and developing *better* news recommender systems requires a research approach which focuses on what users (humans) want, as its ultimate objective, rather than a mathematical notion of accuracy.

The primary objective of this work is not to critique existing research on deep learning models for news recommender systems, which are excellent works on their own terms. Rather, this work aims to provide an alternative perspective on the way academics and practitioners think about researching and developing deep learning models for recommender systems. Fundamentally, we may categorize two driving forces or research-philosophies for academic research. On the one hand, academic advancements may be driven by drilling deeper and deeper into a specific expertise, e.g. exploring more and more complicated deep learning models to develop slightly more accurate models. This incremental, vertical approach to research lies at the foundation of the majority of academic works, also in the field of deep learning. On the other hand, academic advancements may emerge from combining insights from seemingly distant research fields to arrive at novel ideas and avenues for academic research. In fact, the entire field of machine learning originates from such a collision between two apparently distant research-fields when Warren McCulloch and Walter Pitts formulated a mathematical model that mimicked neurons in the human brain in 1943 [14]. This combination of insights between the fields of calculus, computer science and neurobiology led to the an entire new research field, which we now know as machine learning or artificial intelligence. This work aims to explore this alternative research-philosophy, based on lateral thinking [1], in the context of deep learning models for (news) recommender systems.

This work aims to explore a multidisciplinary, lateral thinking approach for researching and developing deep learning models for (news) recommender systems. In doing so, this work aims to address three pitfalls or "traps" that traditional, vertical research may fall into when researching deep learning models. The "traps" this work aims to address are derived from the "abstraction traps" formulated in the work of Selbst et al. [16]. As Selbst et al. formulates it, these traps arise "because academics consider the machine learning model, the inputs, and the outputs, and abstract away any context that surrounds this system." This work argues that academic research on deep learning models for news recommender systems, such as the works reviewed by Wu et al. [17], may fall into one of the following traps, due to the modular research-approach which abstracts away any context surrounding the deep learning model. Below, we describe the three traps these works may fall into.

Formalism Trap

> *Failure to account for the full meaning of social concepts* [which] *cannot be resolved through mathematical formalisms [16].*

[1] Lateral thinking may be defined as: "the solving of problems by an indirect and creative approach, typically through viewing the problem in a new and unusual light." [15]

As mentioned prior, the majority of deep learning research for news recommender systems aims to achieve improvements in terms of certain accuracy metrics (e.g. those mentioned in Section 8.1). These metrics are used to evaluate whether deep learning models provide good accuracy on training data and good generalizability to unseen data. The underlying assumption that accuracy metrics reflect what users want or desire is taken as given and rarely if ever questioned for validity. The objective should ideally be to develop deep learning models which are capable of providing recommendations that lead to the highest user-satisfaction. Accuracy metrics cannot capture the full meaning of a social concept like user-satisfaction. Hence, this work aims to answer how we may capture user-satisfaction more comprehensively for evaluating deep learning models.

Framing Trap

> *Failure to model the entire system over which a social criterion* [...] *will be enforced [16].*

As computer scientists, we are taught to adopt a modular research approach, which clearly frames our inputs, outputs and objectives. In practice this generally means developing a set of deep learning models, which we compare in similar conditions (same dataset) on a set of similar objectives (accuracy metrics). The problem with this approach is that deep learning models rarely operate in such a vacuum in the real world. Recommender systems are usually build up of multiple components, i.e. several different deep learning models and additional methods or algorithms work in conjunction to form a recommender system, as will be discussed in Section 2.2. The result of adopting a modular approach is that deep learning models are validated in solitude, but their performance is rarely evaluated in context of the entire system. Therefore, this work attempts to steer away from this by explicitly adopting a broad view of news recommender systems. In addition, by framing research such that performance of models is evaluated by accuracy, academic works may overlook the fact that deep learning models may exhibit other qualities, which may be more effective in achieving *better* news recommender systems. That is, researchers are eager to claim that one model outperforms another if it provides slightly more accurate recommendations, while neglecting other aspects, such as the diversity of recommendations, which may be more impactful on the user-experience than the difference in accuracy.

Solutionism Trap

> *Failure to recognize the possibility that the best solution to a problem may not involve technology [16].*

The majority of deep learning research is aimed at incrementally improving accuracy metrics, each time slightly improving performance compared to previous methods. Such an approach fails to recognize that there are other qualities or aspects of a (news) recommender system, other than the performance of the deep learning models, which may contribute to the desired outcome. For example, Swearingen et al. [12] showed that recommendations were perceived as more useful by users if more information was provided about the items (e.g. an item description or a review by another user), compared to the same recommendations without additional information. Thus, there may be alternative methods of achieving *better* news recommender systems rather than slightly improving the accuracy performance of the deep learning models used.

The objective of this work is to explore a human-centered approach to researching deep learning models for news recommender systems, instead of being solely focused on accuracy metrics. In addition, this work aims to place deep learning models for news recommender systems in their broader context to reveal opportunities for new insights and research avenues on the crossover between deep learning and their broader context. By adopting such a holistic, multidisciplinary approach, which relies on lateral thinking, rather than a traditional, vertical research-approach, we attempt to address the aforementioned abstraction traps. In doing so, this work attempts to take a step towards a news recommender system, which is able to provide truly satisfying recommendations for users, using deep learning models. To conclude, the objective of this work may be capture as an attempt to answer the following main research question:

RQ How can we move towards a human-centered research- and design-philosophy for developing, implementing and evaluating deep learning models for news recommender systems?

Essentially, our aspiration is to explore how existing deep learning research can contribute more effectively to the development of large-scale news recommender systems, such that recommendations are perceived as more satisfying by users. In its attempt to move towards human-centered deep learning this work explicitly takes a broad perspective of the context in which these deep learning models operate. Nonetheless, this work is realized under certain assumptions to somewhat delineate the scope of our research. First of all, this work omits traditional algorithms or models for (news) recommendations and focuses on state-of-the-art deep learning models, because deep learning forms the predominant method for news recommendation, as will be further contended in Section 2.1. Second, we assume that advancing the field of deep learning for news recommendations requires actionable insights. Accuracy metrics might hold such an established position in deep learning research because they provide clear and precise insights in the performance of models. If we want to move towards deep learning models which aim to provide satisfying recommendations rather than just accurate recommendations, we will need actionable insights to be able to steer research in the right direction. Finally, this work is accomplished under the assumption or belief that placing deep learning models in their broader context will reveal new opportunities for academic research. Based on these assumptions, we formulate three sub-questions that this work aims to address:

SQ1. What is the state-of-the-art in deep learning models for news recommender systems?

SQ2. How can we better understand and measure whether news recommender systems provide recommendations that users want?

SQ3. What is the context in which deep learning models for news recommender systems operate?

1.1 Contributions

The main contributions of this thesis are the following:

1. This thesis provides a discussion of the state-of-the-art in deep learning for news recommender systems and provides a systemic view to promote research into interconnected components, in an attempt to steer away from the framing trap.

2. This work attempts to facilitate a shift towards a human-centered approach to deep learning research for news recommender systems. By discussing the objectives of users and the qualities of news-items and recommender systems that may be perceive as useful, this work may contribute to a better understanding of what recommendations users truly want.

3. Beyond-accuracy evaluation metrics are discussed and formulated in an attempt to provide a more comprehensive set of measures to determine the quality of news recommendations, which is more in line with what users want than accuracy metrics alone. Thus, we aim to take a stride towards tackling the formalism trap.

4. This thesis places news recommender systems in their application, organizational and societal context to reveal insights and avenues for further interdisciplinary research. In doing so, this thesis aims to take a step towards overcoming the framing trap and aims to promote interdisciplinary research on the crossover between news recommender systems and their surrounding context.

5. A case study on the MIND dataset [5] is presented which implements an end-to-end news recommender system based on various deep learning models. In addition, several experiments are conducted to provide practical recommendations for researchers and practitioners to move towards a human-centered approach for researching such recommender systems.

1.2 Organization

The remainder of this thesis is structured in an attempt to answer the three sub-questions and to arrive and a conclusion about how the field of deep learning can move towards a human-centered approach. Essentially, this thesis is structured to take an increasingly broadened view of deep learning models for news recommender systems. First, Section 2 examines the current state-of-the-art in deep learning models for news recommendations and establishes a systemic framework in which deep learning models operate. Next, Section 3 explores how we may form a better understanding of what news recommendations users want and how this can be measured to provide actionable insights for academic research. Subsequently, Section 4 explores the broader context in which deep learning models and new recommender systems reside, including its application context, organizational context, and societal context. The first three sections of this work are thus structured to provide an increasingly broad perspective on deep learning for news recommender systems, which may be illustrated by the pyramid in Figure 1.1.

The first three sections of this work form a strictly theoretical approach in an attempt to answer sub-questions SQ1, SQ2 and SQ3. The latter part of this work, instead, takes a practical approach to investigating human-centered deep learning models for news recommender systems in Section 5. This is done in the form of a case study, which is performed

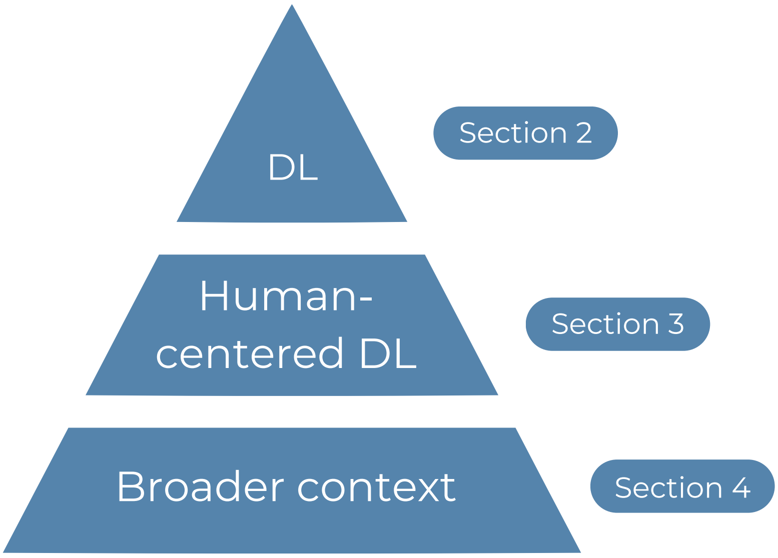
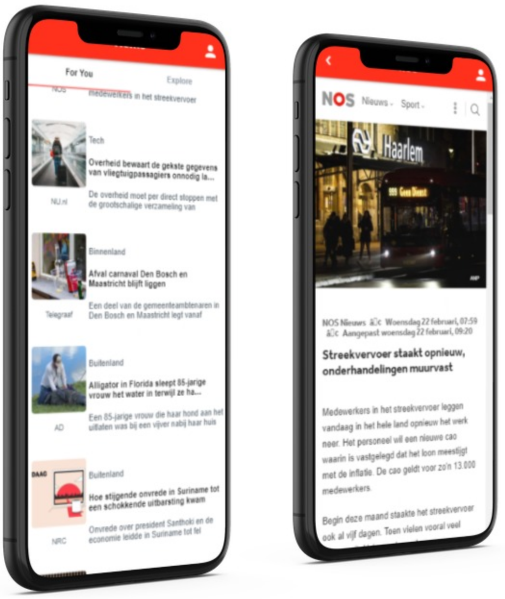


Figure 1.1: Structure of this thesis

on the MIND dataset [5]. Based on the findings of Section 2 this work implements state-of-the-art deep learning models for news recommendations and places these models in an operational end-to-end multi-stage recommender system. Next, the performance of these deep learning models is evaluated on the basis of the findings in Section 3. Ideally, the end-to-end news recommender system based on various deep learning models developed in the case study is examined in its broader context, by implementing it in an actual application and having users interact with the system. However, this exceeds the possibilities and resources for this work, hence the system and models are evaluated on an offline dataset. Limitations like these and directions for further research are discussed in Section 6. Finally, Section 7 summarizes this work and presents the conclusions.

1.3 Preliminaries

Due to the broad, multidisciplinary nature of this work, it may be worthwhile to establish a common understanding of the type of news recommender systems this work refers to without diving into the technical details of the underlying deep learning models. First, let us explicitly mention several news systems this work does not address. This work does not address any information systems used by journalists to explore potential news-stories. This work also does not refer to search engines for news articles. Instead, this work focuses on news recommender systems that form the fundamental basis of a consumer application. That is, our main focus in on large-scale news aggregators that use recommender systems as their main method of providing news to users, e.g. Google News [1] or Feedly [2]. This excludes traditional media outlets which may merely use a news recommender system as a feature besides their traditional methods of selecting and presenting news articles to users, although the findings in this work may still be applicable to some extent. This work has adopted a fictitious use case to showcase the type of recommender systems this work addresses, which is a mobile application with a single, scrollable timeline (not necessarily chronological) that recommends news articles from multiple news sources (see Fig. 1.2). The assumption is that the timeline of such a mobile application is entirely controlled by a news recommender system that relies on deep learning methods, without any human redacting.



medewerkers in het streekvervoer

Overheid bewaart de gekste gegevens van vliegtuigpassagiers onnodig la...

De overheid moet per direct stoppen met de grootschalige verzameling van

Afval carnaval Den Bosch en Maastricht blijft liggen

Een deel van de gemeenteambtenaren in Den Bosch en Maastricht legt vanaf

Alligator in Florida sleept 85-jarige vrouw het water in terwijl ze ha...

Een 85-jarige vrouw die haar hond aan het uitlaten was bij een vijver nabij haar huis

Hoe stijgende onvrede in Suriname tot een schokkende uitbarsting kwam

Onvrede over president Santhoki en de economie leidde in Suriname tot fel

NOS Nieuws â c Woensdag 22 februari, 07:59 â c Aangepast woensdag 22 februari, 09:20

Streekvervoer staakt opnieuw, onderhandelingen muurvast

Medewerkers in het streekvervoer leggen vandaag in het hele land opnieuw het werk neer. Het personeel wil een nieuwe cao waarin is vastgelegd dat het loon meestijgt met de inflatie. De cao geldt voor zo'n 13.000 medewerkers.

Begin deze maand staakte het streekvervoer ook al vijf dagen. Toen vielen vooral veel

Figure 1.2: Mockup of news recommender system application

[1] www.news.google.com

[2] www.feedly.com

2

Deep Learning Framework

In this section, we first investigate the state-of-the-art in deep learning models for recommender systems in Section 2.1 to examine research-question SQ1. Besides providing a conceptual overview of current deep learning approaches for recommender systems, Section 2.1.1 also provides a mathematical framework. Next, Section 2.2 discusses a multi-stage modeling framework, consisting of a retrieval and a ranking stage, which is often adopted in real-world applications of deep learning for recommender systems. Section 2.2.1 and Section 2.2.2 further explore how deep learning models can be adopted for the ranking and retrieval stage. Retrieval and ranking form the core components of effective multi-stage recommender systems, therefore, this work examines both in conjunction. As this work attempts to take a broad perspective of news recommender systems, Section 2.3 discusses the interplay between both components.

2.1 Two-tower Deep Learning Models

Traditionally, recommender systems can be categorized as collaborative filtering, content-based filtering or a hybrid approach [18], but recently deep learning methods have become a popular method for (news) recommendations [19]. Raza and Ding [19] identified a clear rising trend in the use of deep learning algorithms for news recommender systems in their survey. The same rising trend is shown in Figure 2.1, which shows the number of publications in recent years that mention both "personalized news recommendations" and "deep learning" in the title or abstract. In addition, a recent survey by Wu et al. [17] also indicates that deep learning methods are the predominant method for news recommendations. Furthermore, numerous researchers have demonstrated positive results using deep learning methods [5; 17; 20; 21; 22; 23; 24; 25; 26; 27; 28]. Thus, deep learning seems to remain the predominant method for news recommender systems in the near future. Hence, this work will limit its discussion of models for news recommendation to deep learning methods. Readers interested in a review of more traditional methods for news recommendations may turn to one of the many surveys published in the past years [19; 29; 30; 31; 32; 33].

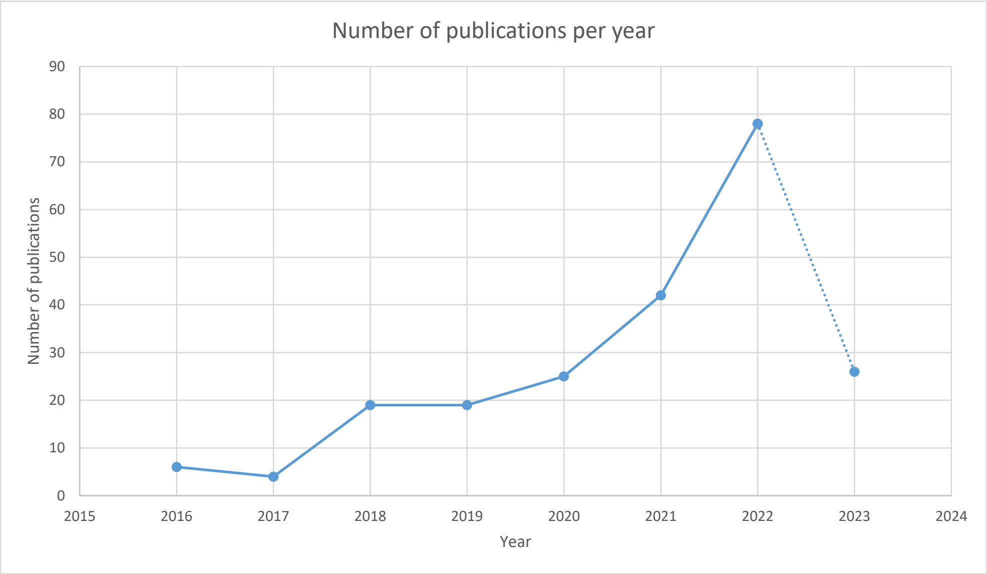


Figure 2.1: Number of publications containing both "personalized news recommendations" and "deep learning" in the title or abstract.

Most deep learning models for news recommendations proposed in recent years follow a common two-tower architecture [17], which can be represented by three core components: a user module (user encoder), a candidate item module (news encoder) and a prediction module, see Figure 2.2. The user module and candidate item module are two independent models, which create a representation for the user and candidate items, respectively. Next, the prediction module uses the representations created by the user module and candidate item module to make a prediction whether the candidate item matches the user. The two-tower architecture has also been shown to provide promising results outside the domain of news recommendations, for example to recommend mobile apps [34], movies [35; 36], music [35], images [36], books [37], and videos [38; 39]. Below, the three modules of the two-tower architecture will be briefly discussed.

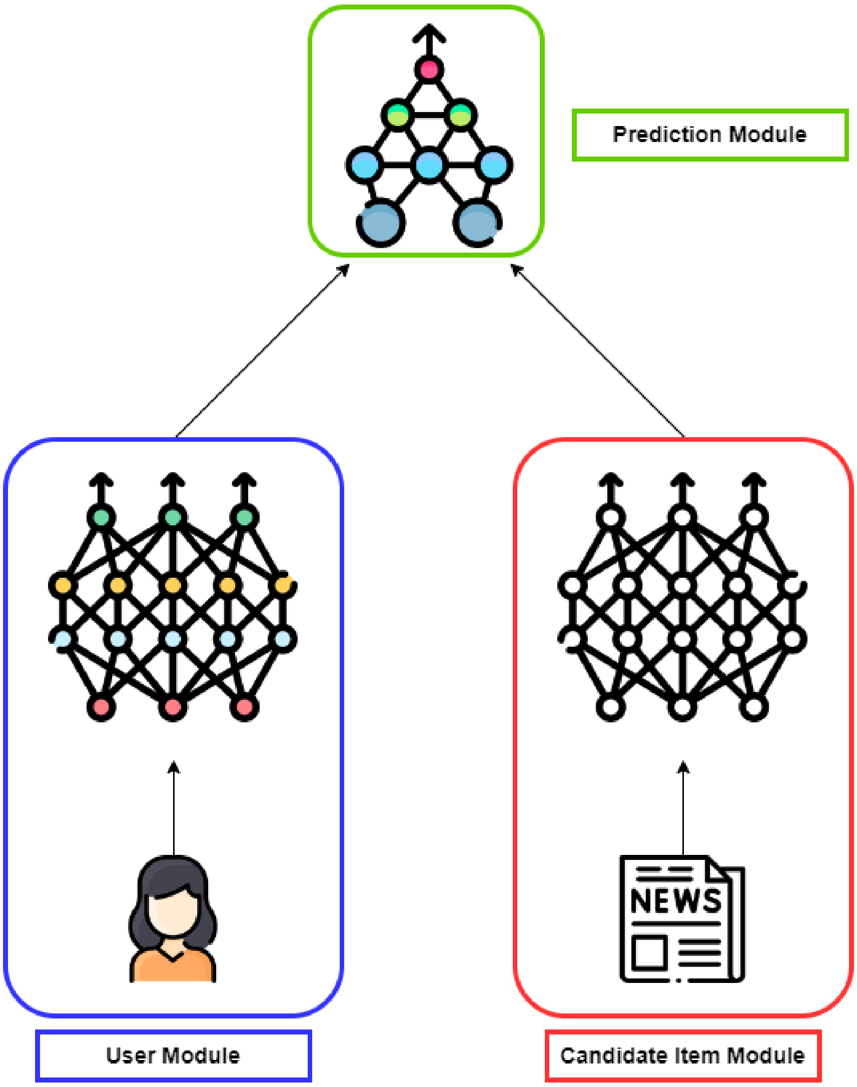


Figure 2.2: Framework of the core components in new recommendation models.

Candidate Item Module (News Encoder) The candidate item module, or news encoder, aims to create a meaningful representation of a news item, based on its characteristics and content. Wu et al. [17] distinguish two approaches to modeling news items: feature-based and deep-learning based. Feature-based modeling approaches rely on hand-engineered features, such as TF-IDF, entities, or topic-distribution (LDA) [17]. Deep-learning approaches, on the other hand, use NLP techniques and neural networks to automatically learn news representations. Feature-based approaches usually require more effort and domain knowledge, and are usually not capable of capturing the sementic meaning of news items as well as deep-learning approaches [17]. Therefore, researchers seem to shift towards deep-learning based approaches.

User Module (User Encoder) The user module, or user encoder, attempts to capture the personal interests and preferences of users. Some of the earlier methods represent users by their ID, but such methods suffer from data sparsity and are inherently limited [17]. Most methods, however, attempt to model users' preferences based on past behaviour, such as the news articles a user has clicked on in the past. In such a case, the semantic meaning of historical news items can be captured in a similar way as done by the candidate item module for candidate news articles. Next, additional neural layers may be applied to capture common interests throughout the historical news articles. Some works incorporate additional contextual features to model the users, such as demographics or location [17].

Prediction Module The prediction module combines the user and news embeddings to determine which news articles are relevant for the user. Several methods determine the relevance of news items for a user based on the similarity in their representations, for example Wu et al. [21] determine the similarity between users and news items by computing the inner product of the two embeddings. In such a case, the inner product can be considered a point-wise prediction of the relevance of a news item for the given user. In case the users are modeled based on news articles which they clicked on in the past, the inner product can also be seen as a prediction of the click-probability, i.e. the probability that the user will click on the news article. This prediction can subsequently be used to rank multiple news items, by sorting the news items by their relevance score or click-probability. Some researchers propose other methods of determining the relevance of news items for users in the prediction module, for example Wu et al. [40] predict the click-probability, probability of finishing the article and the dwell time. In addition, researchers have shown that the prediction module can also be transformed to accommodate multi-task learning, for example Liu et al. [41] introduce a prediction module which predicts the relevance of news items, as well as the category and popularity of the news item.

2.1.1 Mathematical Formulation

The general approach for news recommendations, following the two-tower architecture, can be formulated as follows. Suppose we have a user $u \in U$, where U is the set of all users, and news item $v \in V$, where V is the set of all news items. Our objective is to determine the relevance of a news item v for a particular user u, e.g. if we adopt a click-probability prediction module our objective is to predict whether user u will click on news item v.

Each news item can be represented by an M-dimensional feature vector $\{v_i\}_{i=1}^{M}$, where $v_i \in \mathcal{V}$ may be a wide variety of features, such as the ID, category, title, or content of the news item. Next, each news item is mapped to a k-dimensional embedding space:

$$\psi(v) : V \times \mathbb{R}^d \to \mathbb{R}^k \tag{2.1}$$

where the model ψ represents the candidate item module, which creates an embedding ψ_v for v.

Each user u can also be represented by a set of N features $\{u_i\}_{i=1}^{N}$. A common approach is to represent a user based on the sequence of news items the user has interacted with in the past, i.e. the user u is represented by $H_u = \{v_1^h, v_2^h, ..., v_N^h\}$, where H_u is the sequence of the N historic news items for user u. The set of features $\{u_i\}_{i=1}^{N}$ may also contain additional contextual factors, such as the time, location, or device. Similar to candidate news items, each user u is mapped to a k-dimensional embedding space by a parameterized model:

$$\phi(u) : U \times \mathbb{R}^d \to \mathbb{R}^k \tag{2.2}$$

where the model $\phi(u)$ represents the user module, which creates an embedding ϕ_u for u.

The news encoder $\psi(v)$ and user encoder $\phi(u)$ map news items and users to a shared k-dimensional space, such that if we consider a user u, who prefers item a over item b, the models ϕ and ψ create an embedding ϕ_u for u and ψ_a and ψ_b for a and b, for which we have:

$$sim(\phi_u, \psi_a) > sim(\phi_u, \psi_b) \tag{2.3}$$

where $sim(\cdot)$ is a certain similarity function, e.g. cosine similarity. In addition, a relevance score (e.g. click-probability) may be calculated by combining the user and item representation, which is usually computed as the inner product of the two embeddings:

$$\hat{y} = \langle \phi_u, \psi_v \rangle \tag{2.4}$$

The parameters θ of the two-tower model can be learned from a training set of T examples, denoted by:

$$\mathcal{T} := \{(u_i, v_i, q_i)\}_{i=1}^{T}$$

where (u_i, v_i) is a pair of a user u_i and item v_i, and $q_i \in \mathbb{R}$ the associated reward for each pair [39]. The reward q_i can be an explicit rating of a user u towards item v_i or an implicit feedback metric, such as whether the user has or has not interacted with the item, in that case $q_i = [0, 1]$ for all items. But q_i can also be extended to capture various degrees of user engagement with certain items. For example, q_i can be the fraction of the news item read by a user.

The task of selecting the appropriate items to recommend from a larger set of D items, can be treated as a multi-class classification problem, where the likelihood of recommending item v from a corpus of V items is formulated by the softmax probability function:

$$P(v_i|u_j) = \frac{e^{s(v_i,u_j)}}{\sum_{i\in[V]} e^{s(v_i,u_j)}} \tag{2.5}$$

where $s(v_i, u_j) = \hat{y}_{(i,j)}$ denotes the logits predicted by the full model for item v_i and user u_j [34; 36; 39; 42]. Incorporating the reward q_i results in a weighted loss function:

$$L_T(\theta) = -\frac{1}{T} \sum_{i\in[T]} q_i \cdot log(P(v_j|u_i;\theta)) \tag{2.6}$$

When V is very large, it is not feasible to compute the loss function over all items in the corpus. Instead, the loss function may be evaluated over a batch of B pairs $\{(u_i, v_i, q_i)\}_{i=1}^{B}$. For each $i \in [B]$ we may calculate the batch-wise cross-entropy loss as:

$$L_B(\theta) = -\frac{1}{B} \sum_{i\in[B]} q_i \cdot log(P(v_j|u_i;\theta)) \tag{2.7}$$

Finally, the parameters of the full model can be updating the parameters using a gradient descent algorithm, e.g. Stochastic Gradient Descent (SGD), with a learning rate γ:

$$\theta \leftarrow \theta - \gamma \nabla L_B(\theta) \tag{2.8}$$

The mathematical problem formulation as described here, seems to be the predominant approach in academic works at the time of writing. Nonetheless, numerous researchers have proposed deviations in their works. Obviously, there can be many deviations in the features used and modeling techniques applied to create embeddings for users and items. Furthermore, we already established that the reward q_i may impact the performance of the recommender system, by changing q_i such that it captures various degrees of user engagement. In addition, there seems to be deviations in the way negative samples are incorporated in the loss function [17]. In addition, certain researchers alter the loss function by incorporating secondary objectives, such as time-decay to promote recency [43], a preference regularizer [26], or a sentiment-based regularizer [44]. To the best of my knowledge, there is no comparative study available that investigates the impact of variations in the loss-function. In addition, the majority of publications evaluate the performance of methods only against accuracy metrics, neglecting other aspects that may be of importance for improved user-satisfaction. Hence, there is a lack of evidence whether proposed methods truly improve user-satisfaction.

2.2 Multi-stage Recommender Systems

In commercial settings, recommender systems are usually build up of multiple components (multi-stage RSs), to improve the scalability and latency of the system. The two-tower deep learning models discussed in Section 2.1 offer a high quality of recommendations, but are computationally intensive. Therefore, it is impractical to evaluate all items in the corpus ad hoc, i.e. when a user makes a request for a recommendation. Instead, recommending items to users is split up in multiple stages, which together form a pipeline to select relevant items from a large collections of items. There are different blueprints for multi-stage recommender systems in circulation, but in its essence each system consists of at least two stages: a candidate retrieval stage followed by a ranking stage [45], see Figure 2.3. The two-stage system design has previously been adopted by *YouTube* to recommend videos [38; 39; 46], by *LinkedIn* to recommend jobs [47] and by *Pinterest* to recommend social posts [48]. In this section, we first discuss how we may use deep learning models to approach the ranking stage in Section 2.2.1. Next, we investigate the candidate retrieval stage in Section 2.2.2.

recommender systems may be split up over multiple stages. In addition, we evaluate the benefits and challenges that follow from a multi-stage recommender system design.

2.2.1 Ranking Stage

The two-tower deep learning models discussed in Section 2.1 can readily be applied to rank a set of candidate items for a particular user. Since, two-tower deep learning models are capable of accurately modeling users' preferences, the predictions of such a model can be used to rank items by their utility for users. Suppose we have a two-tower deep learning model $s(v_i, u_j)$, which predicts the relevance of item v_j for user u_i, i.e. $\hat{y}_{(i,j)}$, for all items in a set of candidate items C. Then, the predictions of the two-tower deep learning model are readily converted into a ranking model, by sorting items based on the predicted utility:

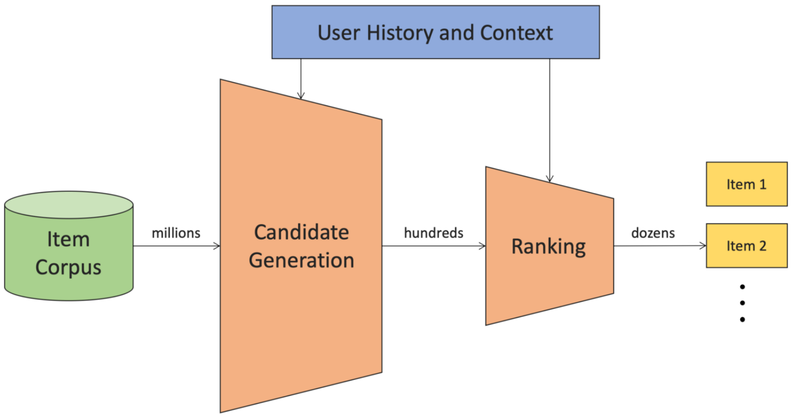


Figure 2.3: Two-stage recommender system (Image by [6])

$$R_u = \sum_{k=1}^{|C|} \arg\max_{c_k \in C} s(c_k, u_j) \tag{2.9}$$

with R_u the ranked list of recommendations for user u and c_k the k-th candidate item in the set of candidates C. Thus, it is possible to accurately rank a set of candidate items, provided that the set of candidate items is of a reasonable size.

Two-tower deep learning models are computationally intensive, which makes it impractical to select relevant items from the entire corpus of items ad hoc. Instead, candidate retrieval methods are adopted in real-world applications to rapidly select a small subset of items, in the order of a few hundred items, as candidates from the entire corpus, which can be in the order of millions or billions. Thereafter, the subset of candidate items can be evaluated by a two-tower deep learning model to select and rank the very best items, usually in the order of dozens, from the set of candidates. Below, the candidate retrieval stage in multi-stage recommender systems is briefly discussed.

2.2.2 Candidate Retrieval Stage

The candidate retrieval (or candidate generation) stage is often the first stage of a recommender system and its aim is to find an adequate subset of potentially relevant candidate items from a large collection of items. In the candidate retrieval stage, a subset of items in the order of hundreds is selected from a collection of items in the order of millions or billions. Because this stage deals with an immense amount of data, the methods used for candidate retrieval are generally light-weight models optimized for efficiency over accuracy. Many of the approaches and methods used in candidate retrieval for recommender systems find their origin in information retrieval. For an extensive introduction to information retrieval readers may turn to Manning's textbook on information retrieval [49].

The candidate retrieval stage has to satisfy two important properties. First, given the large corpus of items, candidate retrieval needs to be scalable to ensure tolerable latency and computational effort [50]. Second, the retrieved candidates must be of sufficiently high quality, which requires precisely modeling interactions of users' and items [50]. Conventional methods represent users and items by constructing similar features for both, e.g. keywords, categorical tags or LDA vectors. This allows for retrieving candidate items for a user based on the feature similarity. For this operation, similarity search, index structures like Locality Sensitive Hashing (LSH) and KNN graphs can be constructed in advance, which significantly increases efficiency. Conventional methods are, however, limited in their capability to accurately represent users and items, due to modeling items and users through hand-engineered features. More recently, deep learning-based systems, such as Wide&Deep [51], Deep&Cross [52], DeepFM [53] and xDeepFM [54], were proposed to capture user interest with high precision. However, in these cases users and items are no longer represented by similar feature representations. Therefore, embedding-based retrieval methods were developed to leverage the efficiency and scalability of conventional methods and the accuracy of deep-learning approaches [7; 38; 39; 50; 55].

Here I will briefly formulate a general approach to embedding-based similarity search retrieval methods using deep-learning models. Embedding-based similarity search methods leverage the same principles underlying the two-tower deep learning methods discussed in Section 2.1. We already established that two-tower deep learning models are able represent users and items accurately in a shared latent space, such that given a user u who prefers item a over item b, we have:

$$sim(\phi_u, \psi_a) > sim(\phi_u, \psi_b) \tag{2.10}$$

where $sim(\cdot)$ is a certain similarity function, e.g. cosine similarity, and ϕ_u, ψ_a and ψ_b are the representations of the user, item a and item b, respectively. Thus, candidate items can be retrieved directly using similarity search methods (or approximate nearest neighbor ANN search), such that items can be indexed for efficient retrieval. This makes it possible to store user and item representations in an indexed vector database (e.g. Faiss [56]) and utilize similarity search methods like KNN-graphs [57; 58].

2.3 A Systemic Perspective

The two fundamental stages, retrieval and ranking, together form an end-to-end news recommender system. Nonetheless, there are only a handful of publications available that investigate these two stages in conjunction, such as [39; 55; 59]. To the best of my knowledge, there is no work available that evaluates both stages in conjunction for news recommendations. From an academic perspective, it may make sense to break up research in individual components, since this makes it straightforward to formulate and frame the problems. In addition, there are numerous benchmark datasets available for each individual component in isolation. However, if we adopt a broader systemic perspective on news recommender systems and the role deep learning models play in such a system, as this work attempts to do, this seems as an opportunity for exploring the interplay between both stages. To quote Liu et al. [50]: *"It is apparent that the candidate retrieval operation severely affects the overall recommendation quality, whose performance has to be optimized within the integral system."* In this section, we will explore the possible interplay between the candidate retrieval stage and ranking stage from a theoretical perspective, which may reveal insights or opportunities for further research aimed at the entire system shown in Figure 2.3.

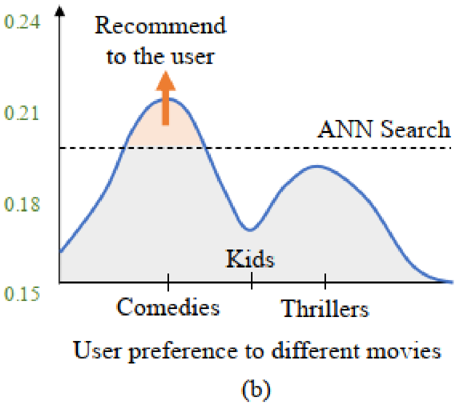


Figure 2.4: Similarity-search for users with multiple interests (Image by [7]).

Candidate retrieval methods may play a principal role in the items that are being recommended to users downstream. From a theoretical point of view, it seems likely that the results of the ranking stage in a multi-stage news recommender system, rely heavily on the candidate-items it is receiving as input from the candidate retrieval stage. Therefore, candidate retrieval methods may play a key role in ensuring novelty, diversity and serendipity in the final recommendations, but there are still open-ended questions that require further academic investigation. First of all, this work recommends researchers and practitioners to investigate the magnitude with which the candidate retrieval stage impacts the ranking stage. Secondly, assuming the candidate retrieval stage impacts the latter ranking stage, it may be worthwhile to investigate how to optimize the retrieval stage for desirable results downstream. Furthermore, embedding-based similarity search methods offer high flexibility in feature representations, which makes it possible to combine multiple retrieval methods to improve novelty or diversity. Therefore, it may be possible to combine multiple retrieval methods which each capture a different aspect of items and users' preferences in order to improve performance and diversity.

On the contrary, candidate retrieval may hinder diversity, novelty and serendipity in recommendations, precisely because these methods rely on similarity-search methods [7; 9]. Similarity-search methods only greedily retrieve the top-scored items, which may all be very similar items. In most applications, including the news domain, users usually have a diverse set of interests, but this may be overlooked by similarity-search methods. This problem is illustrated in Figure 2.4, which shows the calculated preference score for a number of movies for a particular user (blue line) and the items that are retrieved by similarity-search methods (items above the dotted line labeled "ANN Search"). In this case, this particular user has this highest interest in comedies, which are indeed retrieved using approximate nearest neighbor (ANN) search. However, the user also shows an interest in thrillers, but these movies are not selected by the retrieval method, because the method only retrieves the top-scored items. From a user perspective it may be more desirable to retrieve some comedy movies and some thrillers to improve diversity and the overall user-experience.

Some researchers have proposed methods to improve the diversity in embedding-based retrieval methods. Zhang et al. [7] propose a divide and conquer strategy, which retrieves items from several clusters in the item space (i.e. from different peaks in Figure 2.4). El-Kishky et al. [9] propose kNN-Embed, an approach which transforms a single user embedding into a smoothed mixture of learned item clusters to capture distinct user interests. Zhang et al. [7] report significant improvements in user engagement in live A/B experiments using their divide and conquer retrieval strategy. El-Kishky et al. [9] report both an improvement in recall and diversity in offline evaluation using their kNN-Embed method. Furthermore, since embedding-based similarity search can be performed in near real-time, it is possible to incorporate contextual information, such as the time, location or user-device, which may further improve accuracy of recommendations.

To conclude, adopting a systemic perspective of news recommender systems may reveal new opportunities for improving recommendation performance. By highlighting the connection between both stages, as illustrated in Figure 2.3, this work aims to promote academic research investigating such systems in their entirety. Throughout this section discussing the interplay between both the retrieval and ranking stage, we have mentioned several qualities of retrieval and ranking models, such as diversity, novelty and serendipity. However, most publications on retrieval and ranking methods omit such qualities in their research and only evaluate performance using accuracy metrics. In the following section, we will further investigate these additional qualities in more detail.

3

Human-Centered Deep Learning

The majority of deep learning research for news recommender systems is guided by accuracy metrics. Using accuracy metrics as the sole measure for performance fails to recognize that user-satisfaction or the perceived usefulness of recommendations cannot be captured by a single mathematical formalism. In this section we explore what makes a news recommender system *better* and how we can form a better understanding of what users actually want and how this may be quantified in an attempt to answer SQ2. We may take several approaches in trying to uncover what users want from a news recommender system and how deep learning research may capitalize on this. First, we take a human perspective on news recommendations in an attempt to uncover what users want in Section 3.1. This includes the goals users are trying to achieve when using a news recommender system in Section 3.1.1, and the qualities of recommender systems and news items that users perceive as desirable in Section 3.1.2. In the latter part of this section, we attempt to provide insights into ways deep learning researchers can shift towards a more comprehensive set of measures for evaluating the performance of news recommender systems in Section 3.2.

3.1 A Human Perspective

3.1.1 Goals

Herlocker et al. [60] identify ten tasks or goals users may wish to achieve with the help of a recommender system. In the context of news recommendations, the objective of users will likely be what Herlocker et al. [60] categorizes as “finding some good items”, i.e. users will likely want to read some (not necessarily all) news articles that meet their personal preferences or interests. Since users may read several news articles in a sequence, it may be fruitful to attempt to recommend a sequence of news articles which may be more pleasing as a whole, as exemplified by the work of Abdollahpouri et al. [61]. In addition, Herlocker et al. [60] note that in certain recommender system applications users may occasionally only be browsing, without actively interacting with recommendations. The same may potentially be observed in the context of news recommendations, where certain users may merely wish to scan the headlines. Identifying the task users attempt to perform with a news recommender system may allow for a system design that tailors towards that task.

Regardless of whether users are reading a single article, sequence of articles or scanning articles, the ulterior motive for reading news is much more complex and encompasses desires such as being informed, socially engaged, and amused [62]. According to Schrøder [62] users want to read news because they perceive it as relevant to their personal lives, such as their family, workplace or local community. Furthermore, she argues that one of the reasons users want to read news is to be more socially engaged. People may read stories because they belief others around them might take an interest in it. Bengtsson [63] also reveals some insights into reasons why people want to read news, such as that people prefer articles which they perceive as useful for themselves in the near future and for positioning themselves in the world.

Here we presented two ways of thinking about the goals users wish to achieve with a news recommender system. The first, rather practical in terms of the how users interact with the items, and the latter from a broader social perspective. We contend that unveiling such goals may yield a better understanding of the purpose of developing deep leaning models and news recommender systems. As Herlocker et al. put it: "[The] *performance of a recommender system must be evaluated with respect to specific user tasks and the domain context."*

3.1.2 Desirable Qualities

Understanding qualities which are perceived as useful or desirable may unveil new insights in developing better news recommender systems. Both the qualities of the system itself and the items being recommended play a role in how users perceive the recommendations. Sundar [64] provides a comprehensive examination of the qualities of news stories that impact people's perception of the news items, based on interviews and questionnaires. Based on the responses of participants, Sundar categorizes the perceived qualities of news articles in four distinctive categories: credibility, liking, quality and representativeness. Each category was associated by several qualities, which respondents had to rank. For example, credibility was expressed in terms of how biased, fair and objective respondents perceived the news article. For a full list the qualities Sundar identified as most influential for respondents perception of news articles, see Table 1 in his work [64]. Most news recommender systems, however, gather news articles from external sources and, thus, have very little control over the content that is being fed into the system.

Instead of looking at the qualities of individual news items, it may be more fruitful to investigate qualities of the system that affect users perception of the system and the recommendations. Swearingen et al. [12] conducted a user-study involving a questionnaire and interview to investigate the qualities of a recommender system that make whether users perceive recommendations as "good" and "useful". Swearingen et al. pointed out that accuracy of the recommendations correlated highly with perceived usefulness, but also identified several other qualities. First, when users where shown recommendations which that had seen before, this improved users' trust in the system and made them more likely to perceive recommendations as useful. Second, users showed positive responses to new and unexpected recommendations. Furthermore, Swearingen et al. showed that improving the transparency of the system and available information about items improved the perceived usefulness of the system. Pu et al. [8] also conducted user-studies to investigate the qualities of recommender systems that affect users perception of the system. Based on participants responses about the qualities of a recommender system through a questionnaire, Pu et al. constructed a user-centric evaluation framework for recommender systems. This structural model is shown in Figure 3.1. Qualities such as the recommendation accuracy, recommendation novelty and recommendation diversity were shown to have a positive impact in users intention to purchase or use the recommended item, i.e. these qualities had a positive effect on the perceived usefulness of recommendations. It is worth noting that these descriptions of the perceived qualities were not deducted by the researchers, but rather expressed by the respondents themselves. For example, the diversity of recommendations was determined by asking respondents to rate the following question on a 5-point Likert scale: "The items recommended to me are diverse."

3.2 Evaluating System Performance

Designing appropriate evaluation methods is crucial for assessing the performance of recommender systems in the context of human-centered news recommendations. Firstly, evaluation metrics provide a standardized framework to compare and benchmark different retrieval and ranking models, enabling researchers and practitioners to identify the most effective approaches. Secondly, proper evaluation techniques facilitate the evaluation of recommender systems under various conditions, such as different user preferences, item

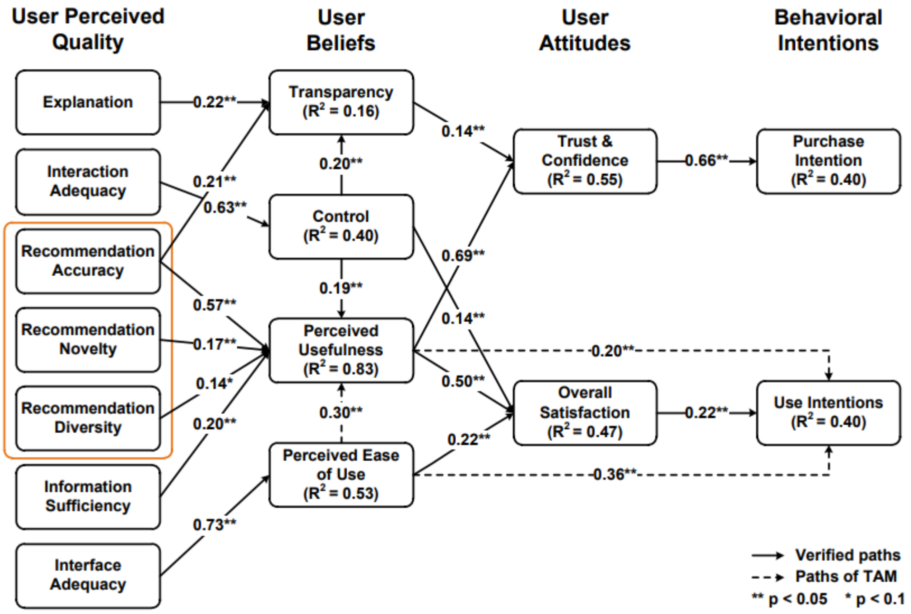


Figure 3.1: User-Centric Evaluation Framework of Pu et al. (Image by [8]).

characteristics, or contexts. This allows for a better understanding of the strengths and limitations of recommend systems in different scenarios. Thirdly, appropriate evaluation methods can be adopted to ensure that the systems keeps operating as expected in production, eliminating the risk of model drift. Ultimately, selecting appropriate evaluation methods leads to a better understanding of recommender systems, ensuring their ability to deliver relevant recommendations and enhancing user satisfaction.

Many researchers have pointed out that evaluating the utility of recommender systems in terms of accuracy may not suffice [1; 4]. Some user studies have indicated that recommendation accuracy may the the most important factor contributing to the utility of recommendations, such as the one conducted by Swearingen and Sinha [12]. However, this does not mean that accuracy is the only factor contributing to the perceived utility of recommendations [4], as we have discussed in Section 3.1. To develop deep learning models using a human-centered approach, researchers need to shift from a purely accuracy-focused approach, to a broader set of metrics. In this section, we will present several metrics proposed in the literature to quantify beyond-accuracy metrics, such as diversity, novelty, recency, serendipity and unexpectedness.

Prior to discussing the various evaluation methods, this section first includes a note on collecting user feedback in recommender systems in Section 3.2.1. The way user feedback is collected in recommender systems may impact to what extent conclusions can be drawn about the preferences of users. Next, Section 3.2.2 discusses methods to evaluate the diversity of recommendations, as diversity may be an important aspect in providing satisfying recommendations for users with distinct interests. In Section 3.2.3 methods to evaluate the novelty and recency of recommendations are presented, since including a small portion of novel items has been shown to have a positive impact on user satisfaction [8; 12]. Finally, in Section 3.2.4 methods to evaluate the serendipity and unexpectedness of recommendations are reviewed. Unexpected recommendations may be more interesting to users and might challenge users to expand their tastes [4]. In this work we focus our attention on diversity, novelty and serendipity. Diversity and novelty are the two qualities of recommendations, besides accuracy, identified by Pu et al. to be indicative of users' perception of the system. In addition, we review serendipity and unexpectedness, because Swearingen et al. showed that users responded more positively to recommender systems that recommended new and unexpected items.

3.2.1 User Feedback

An effective recommender system should be able to adequately model users' preferences, which depends largely on the way users' preferences are elicited. First of all, a distinction can be made between systems that allow users to interact with the system anonymously and systems that require explicit user profiles [19]. The former case being especially challenging, because of the short time-frame and the fact that each user-session is faced with the cold-start problem [65; 66]. Several session-based methods have been proposed in the domain of news recommendations, such as the works of Sottocornola et al. [65] and Trevisiol et al. [66]. These methods are, however, incapable of modeling user preferences over the span of multiple sessions and may be highly biased towards popular items. Secondly, user preferences may be modeled based on either explicit or implicit feedback [19]. Explicit feedback, most commonly ratings, is easily quantifiable, but may be intrusive to the user experience or may be neglected by users [67]. Hence, many recommender systems rely on implicit feedback, such as clicks on items or interaction time, to determine users' preferences [19].

Implicit feedback is widely used to model users' preferences in (news) recommender systems, but the use of implicit data should be carefully considered in the design process. First of all, several academics have pointed out that implicit feedback metrics may not be as accurate as expected [68; 69]. For example, using clicks as a proxy for which news articles are relevant for users may be hindered by the presence of undesirable articles with click-bait titles. Likewise, using the time spend reading as a metric may favour lengthy articles or may be skewed by idle time of users [69]. In addition, implicit feedback often lacks a clear negative signal, such that for many items no feedback is available [10]. Items for which no feedback is available can, therefore, either be irrelevant items users chose to ignore or perfectly relevant items which were simply not discovered by users [10]. This possible bias should, therefore, be carefully taken into account when constructing a recommender system based on implicit feedback data.

3.2.2 Diversity

Diversity, in the context of recommender systems, refers to how dissimilar the items in a list of recommendations are from each other. The informal argument often made in the literature in favour of diversity is that a list of very similar items may quickly bore the user [4]. This intuitive argument is supported by research which has shown that users will actively attempt to improve the diversity of recommendations by choosing less-preferred items [2; 3]. To measure the diversity of a list of items we need some notion of what it means for two items to be similar or dissimilar. In fact, if we have an arbitrary function:

$$sim(r_i, r_j),$$

which returns a similarity score for two recommended items r_i and r_j, than we can also calculate a similarity score for a set of R recommended items (Note that a score for the similarity of a set of items can simultaneously be used as a measure for how diverse the set of items is, i.e. the lower the similarity score, the higher the diversity). A popular way to calculate the similarity over a set of items is to use the *Intra-List Similarity* (ILS) metric introduced by Ziegler et al. [2]:

$$ILS(R) = \frac{\sum_{r_i \in R} \sum_{r_j \in R, r_i \neq r_j} sim(r_i, r_j)}{2} \tag{3.1}$$

Today it is more common to replace the denominator by the number of comparisons to report the average pairwise similarity [70]:

$$ILS(R) = \frac{\sum_{r_i \in R} \sum_{r_j \in R, r_i \neq r_j} sim(r_i, r_j)}{((|R|)(|R| - 1))/2} \tag{3.2}$$

Another variation of the ILS-metric was proposed by Sheth et al. [71], who propose to calculate the ILS as an average of the number of recommendations in the list:

$$ILS(R) = 100 \times \frac{\sum_{r_i \in R} \sum_{r_j \in R, r_i \neq r_j} sim(r_i, r_j)}{|R_u|} \tag{3.3}$$

One approach to score the similarity between two items is to calculate a certain distance metric between the latent representations of the two items [11]. Constructing proper latent representations is highly context dependent, which makes it complicated to extrapolate from previous research [72]. In addition, in many applications there is a lack of confirmation whether the similarity between latent representations accurately reflects the human perception of similarity [70]. Zhang et al. [4] overcome this limitation by scoring the similarity between two items based on a Collaborative Filtering memory-based similarity metric (*CosSim*):

$$\textit{CosSim(i,j)} = \frac{\text{\# of users who like both i and j}}{\sqrt{\text{\# of likes for i}} \times \sqrt{\text{\# of likes for j}}} \tag{3.4}$$

This metric determines the similarity between two items based on the proportion of users the two items have in common [4]. The intuition behind the metric is that similar items will probably be liked by similar users. The advantage of this method is that it is domain-independent, which makes it a popular choice for academics [11]. Nevertheless, it is again not guaranteed that this metric coincides with diversity as perceived by users.

There are several other diversity metrics proposed in the literature. Fleder et al. [73] represent diversity in e-commerce product recommendations using the Gini coefficient, but according to Kunaver et al. [72] it is questionable whether this metric can be applied to other domains. Vargas et al. [74] calculate diversity based on a combination between genre coverage (number of different genres present in the list of recommendations) and non-redundancy (genres do not repeat on the list). The limitation of this metric is that it only works for items corresponding to genres and it is still unclear how to include other factors that may contribute to perceived diversity. Another approach is to calculate diversity as part of the nDCG measure, as done by Clarke et al. [75] and Vargas [76], but this requires extensive data to be correctly evaluated. For an comprehensive survey of the diversity metrics proposed in the literature, readers may turn to Kunaver et al. [72]. Kunaver et al. [72] note, however, that very few researchers agree on the metric that should be used for diversity. The metrics proposed by researchers are often highly domain-dependent or presented in such an abstract manner that the details are left to be interpreted [72].

3.2.3 Novelty and Recency

The concept of novelty refers to the ability of a recommender system to introduce users to items they have not previously experienced [4]. The argument for novelty is that recommending items users are very familiar with may be accurate, but also far too obvious and of little help to the user [2]. Both Swearingen and Sinha [12] and Pu and Chen [8] have shown that a small portion of novel items can have a positive impact on user satisfaction. According to Silveira et al. [11], items can be novel to users on three levels. From the highest level perspective (*life level*), an item is novel to a user if the user has never experienced the item in his/her life. This includes any experiences outside the recommender system. It may be possible to evaluate life level novelty of items by surveying users and asking them whether they know the items, but there does not seem to be a metric that quantitatively evaluates this. On a more practical level, items can by novel to users on a *system level* or *recommendation list level.* If an item is novel on the system level, it means that the user has not previously interacted with the item according to the user's history recorded on the system. Novelty on a recommendation list level generally refers to non-redundant items in the list of items recommended in a single session [11].

Avoiding redundant recommendations in the list of items is generally enforced post-ranking, by re-ranking or filtering items. So, although it is important to analyse novelty on a recommendation level, most metrics are designed on a system level in an attempt to improve recommender models. Nakatsuji et al. [77] determine novelty based on the similarity between the items in the list of recommendations and the history of the user:

$$nov(R_u) = \sum_{i \in R_u} \min_{j \in H_u} d(class(i), class(j)) \tag{3.5}$$

The proposed metric is based on a distance metric *d*, which calculates the distance between the classes of two items, based on the number of hops between $class(i)$ and $class(j)$ in a taxonomy. Another measure for novelty is the following metric introduced by Zhou et al. [78]:

$$\overline{novelty} = \frac{1}{|U|} \sum_{u \in U} \sum_{i \in R_u} \frac{\log_2 pop_i}{|R_u|}, \tag{3.6}$$

where pop_i is a popularity score for item i, based for instance on the number of users that have interacted with the item. The metric quantifies to what extent globally "unexplored" items are being recommended on average, based on the assumption that users are more likely to be familiar with globally more popular items. Castells et al. [79] propose an identical novelty-metric although slightly reformulated by first expresing the *generic novelty* of a single item using the Shannon Entropy:

$$novelty(i) = -\log_2 p(i), \tag{3.7}$$

where $p(i)$ represents the probability that item i is chosen by a random user. Next, Castells et al. define the expected novelty ($m(R)$) over a list of recommendations as:

$$m(R) = \sum_{i \in R} p(i|R) novelty(i). \tag{3.8}$$

Assuming that each item i occurs only once in a list of recommendations, we have $p(i|R) = \frac{1}{|R_u|}$ which shows the similarity between Eq. 3.6 and Eq. 3.8. Castells et al., however, make an interesting note that $p(i|R)$ can be replaced by $p(i|R,u)$ by considering to what extent user u has browsed through the list or recommendations R and whether item i is at all relevant to user u.

In many recommender system applications the perceived value of items may be correlated with the lifetime of items. In certain cases, the value of items may deteriorate as the lifespan of the items increases, such as old movies, songs or especially news stories. To what extent a recommender system is able to recommend fresh items is referred to as recency or freshness. Chakraborty et al. [80] propose a measure of recency in the context of recommending news stories, based on the difference between the recommendation time and the time the news story was published. The metric introduced by Chakraborty et al. [80] may be generalized to determine the recency of a recommended item i as follows:

$$\text{recency}_i = \frac{1}{\text{time since } i \text{ entered catalog of RS}}, \tag{3.9}$$

where the time may be in seconds, minutes or hours depending on the application. To effectively compare the recency of multiple items in a list of recommendations, the recency-score ($recency_i$) of each item can be normalized by the most recent item in the list of recommendations:

$$normalized\_recency_i = \frac{recency_i}{\max(recency_i \forall i)} \tag{3.10}$$

3.2.4 Serendipity and Unexpectedness

The term serendipity refers to "lucky findings" or a "satisfying surprise" [11]. Since serendipity is a complex concept, researchers adhere to various definitions of serendipity in the context of recommendations and have proposed various metrics. Zhang et al. [4] define serendipity as the "unusualness" or "surprise" of recommendations. Maksai et al. note that serendipitous recommendations must both be unexpected and useful to the user: "serendipity is the quality of being both unexpected and useful" [81]. Although researchers may debate on the exact definition, the reasoning for serendipity is often similar. A recommendation may be accurate and novel, but hardly surprising [4]. For example, recommending a novel Hollywood movie to a user who usually watches blockbuster movies is hardly surprising, but recommending an arthouse movie is. As Zhang et al. [4] formulate it: "a serendipitous system will challenge users to expand their tastes and hopefully provide more interesting recommendation".

There are several approaches taken by academics to evaluate serendipity. Here we will focus on approaches that evaluate serendipity through a single formula or metric. Two other approaches are evaluation through user surveys and determining serendipity indirectly based on its components, such as novelty and diversity [82]. Ziarani et al. [82] provide a comprehensive survey of serendipity evaluation methods for recommender systems. Based on their survey, they conclude that serendipity can be evaluated as the ratio of the number of useful and unexpected recommendations to the total number of recommendations:

$$Serendipity = \frac{|Unexpected \cap useful|}{|R_u|} \tag{3.11}$$

This provides a simple and intuitive evaluation metric, but lacks guidelines for quantifying when a recommendation is unexpected or useful. The same impractical abstraction is present in several other publications proposing evaluation metrics for serendipity. For example, De Gemmis et al. [83] propose the following evaluation metric:

$$Serendipity = \frac{\sum_{i \in R_u} S(i)}{|R_u|}, \tag{3.12}$$

with

$$S(i) = \begin{cases} 1, & \text{if } i \text{ is serendipitous} \\ 0, & \text{otherwise} \end{cases} \tag{3.13}$$

However, to actually evaluate whether a recommendation is serendipitous or not, the authors rely on a user-study.

Jain and Hasija [84] propose to measure the serendipity of a recommended item i as:

$$S_{count}(i) = \min dist(i, j) \tag{3.14}$$

with

$$dist(i, j) = 1 - \frac{(G_i) \cap (G_j)}{(G_i) \cup (G_j)}, \tag{3.15}$$

where G_i is the set of positively rated serendipitous items and G_j the set of positively rated items. The set of serendipitous items is based on the negative ratings of dissimilar users. Classifying items as serendipitous this way may work in applications where users provide explicit feedback in the form of a rating, but this does not translate to the case where there is only implicit, positive feedback. Namely, if all items without implicit, positive feedback would be regarded as negative feedback, the set of items with negative feedback would likely span the majority of the catalog of items, including unserendipitous items.

Zhang et al. [4] assess serendipity using the cosine-similarity to evaluate the similarity between items in the user's history H_u and the items in the list of recommendations R_u. The more similar the historical items and the recommended items, the less surprising the recommendations are considered to be. That is, lower values indicate more serendipitous recommendations:

$$\overline{Unserendipity} = \sum_{u \in S} \frac{1}{|S||H_u|} \sum_{h \in H_u} \sum_{i \in R_u} \frac{CosSim(i, h)}{|R_u|} \tag{3.16}$$

According to Silveira et al. [11], this metric seems reasonable, since serendipitous recommendations should not be very similar to the user's consumption profile. However, they point out that this metric does not take into consideration whether the recommendations are useful. Hence, the metric might be evaluating novelty or unexpectedness instead.

Unexpectedness may be defined as a "divergence from expected recommendations", which can be considered closely related to or even overlapping with the concept of serendipity [11]. In the literature, there are two approaches proposed to measure unexpectedness: metrics based on a primitive recommender system (PM_u) and metrics not involving a primitive method. The underlying assumption of metrics based on a primitive recommender system, is that the recommendations made by a primitive model are to be expected, whereas unexpectedness will by high for items that cannot be predicted by a primitive model [85]. Ge et al. [86] proposed the following metric for unexpectedness, using a primitive recommender system:

$$unexp(R_u) = R_u - PM_u, \tag{3.17}$$

which was subsequently refined by Adamopoulos et al. [87] to reflect the rate of unexpectedness:

$$unexp(R_u) = \frac{R_u - PM_u}{|R_u|}. \tag{3.18}$$

It is, however, unclear to what extend items recommended by a primitive recommender system may be considered to be expected and items not recommended to be unexpected.

Kaminskas and Bridge [88] proposed a metric for unexpectedness, without using a primitive recommender system, based on a point-wise mutual information function ($PMI(i, j)$). The PMI function calculates the probability of two items i and j to be rated by the same users as follows:

$$PMI(i, j) = \frac{\log_2 \frac{p(i,j)}{p(i)p(j)}}{-\log_2 p(i, j)}, \tag{3.19}$$

where $p(i)$ is the probability of item i to be rated by users. Next, unexpectedness is defined as either of the following two expressions:

$$unexp(R_u) = \sum_{i \in R_u} \sum_{j \in H_u} PMI(i, j) \tag{3.20}$$

$$unexp(R_u) = \sum_{i \in R_u} \max_{j \in R_u} PMI(i, j) \tag{3.21}$$

According to the authors, these metrics may, however, be biased towards rare items, which are always considered to be unexpected.

4

Broader Context

In this section the broader context of news recommender systems is explored to investigate SQ3. Placing news recommender systems in their broader context may reveal new opportunities for academic research in deep learning models for news recommender systems. We can identify several context in which we can investigate news recommender systems, see Figure 4.1. In Section 2 we already established that deep learning models do not operate individually, but rather in a systemic framework. That is, several deep learning models together form a news recommender system. This news recommender system is, subsuquently, part of an application, for example a mobile app as shown in Figure 1.2. This is what we will refer to as the application context of a news recommender system. Furthermore, each news recommender system is part of an organization that operates the system, which forms the organizational context. Next, the people using the application containing the news recommender system form an additional contextual layer. The perspective of users on news recommender systems has already been discussed in Section 3. Finally, each news recommender system exists in a societal context, which forms the final layer in Figure 4.1. In this section, we first explore the application context in Section 4.1. Next, we discuss the organizational context in Section 4.2 and the societal context in 4.3.

4.1 Application Context

Understanding the application context in which a news recommender system operates, may enhance the effectiveness of researching and developing deep learning models. Specialized deep learning techniques may be leveraged for different applications. Furthermore, collaborations between other domains associated with researching news applications and deep

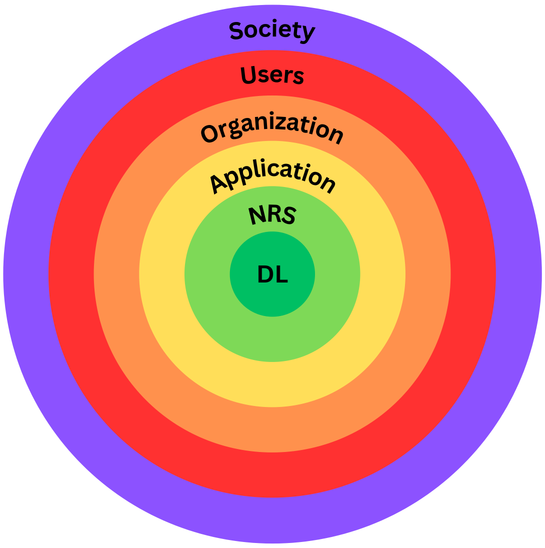


Figure 4.1: Various degrees of context for deep learning models and news recommender systems

learning experts might reveal new opportunities for research, which might otherwise go unnoticed. Let us illustrate this point with an analogy. Consider the short-form video sharing platform YouTube[1], which is powered by a sophisticated recommender system based on deep learning models. YouTube, which is operated by Google, likely has the resources and capabilities to develop some of the best deep learning models for recommending their videos. As such, the home-page of YouTube has likely gotten better and better over time, driven by the increased performance of the underlying deep learning models. Nonetheless, YouTube has recently been overtaken as the dominant video-sharing platform by TikTok[2]. TikTok, like YouTube makes use of sophisticated deep learning models to provide recommendations. However, TikTok also optimized other aspects of their application, such as

[1] `www.youtube.com`

[2] `www.tiktok.com`

being able to swipe through single videos, rather than having to navigate a home-page. TikTok might not have even have more accurate deep learning models than YouTube, instead they found a way to combine their recommender system with a novel interface, which has apperently led to a huge increase in user-satisfaction. Thus, it may be possible to optimize the capabilities of deep learning models or news recommender systems by optimizing it in conjunction with the surrounding application context. We split the discussion of the application context in this section up in two parts. First, we discuss the functionality of the application in Section 4.1.1. Second, we discuss the interface of the application in Section 4.1.2.

4.1.1 Functionality

During the design process and technique selection of a news recommender system, it is vital to identify and describe the required functionality of the recommender system within its application. In Section 3.1.1 we already established that news readers may have different objectives when using a news recommender system. This section builds upon these objectives and discusses in more detail the functionality that users may expect from a news recommenders system. First, the degree of personalization is discussed and the implications this has. Second, the trade-off between user-control and cognitive load is reviewed. Finally, we briefly discuss contextual factors, e.g. time, location, or device, that may be leveraged to develop a context-aware recommender system.

During the design process of a news recommender system, the desired degree of personalization should be established. The degree of personalization may be on an individual level, group level, or system level [89]. In applications where users have distinctive preferences, users will benefit from more personalized recommendations. However, more personalization in news recommendations may also lead to isolating users from diverse viewpoints [90; 91]. In addition, increased personalization may raise users concern about their privacy [92]. Therefore, it is worth evaluating to what degree users benefit from more personalized recommendations. This allows for a well-informed decision in the trade-off between increased personalization and potential drawbacks.

While using a news recommender system, users may expect to be able to assert a certain degree of control over the system, either directly or indirectly. User control may improve the quality of recommendations [93], assert autonomy [94], and increase credibility and trust [8]. Prior to designing a news recommender system, it should be investigated whether users expect to have direct control over the recommendation algorithm(s) or indirect control through a control panel, personal preference profile, or in terms of privacy policies [89]. The desired degree of user control must also be in line with organizational objectives and must also be technically feasible. Furthermore, a balance must be found between the degree of user control and cognitive load, as certain control mechanism proposed in academic literature may be too complex and burdensome for the average user [95].

Contextual factors, such as time, location, or device, may be leveraged to deliver more valuable recommendations [96]. Kille [97] has shown that news-readers behavior differs depending on the time, weekday or device used, which leads to the belief that users' preferences vary depending on the context. In addition, leveraging contextual factors may help alleviate the cold-start problem, as contextual factors are often available even for new users [98]. Therefore, there may be a benefit in adopting a context-aware recommender system. Nonetheless, several researchers have indicated that effectively designing a context-aware news recommendation approach is not a straightforward task [97; 99].

4.1.2 Interface

The interface design of the application containing the news recommender system addresses how, when, and where items are presented to users. Certain interface design choice may be made regardless of the underlying recommendation techniques, however, some aspects do affect the entire system design. First of all, the modeling framework, algorithms and evaluation methods may differ depending on the way items are shown, as well as the number of items shown. Second, the (multi-)modality of news items, including the possible multi-lingual nature of the items impacts the entire modeling framework. In this section we will discuss these two aspects related to interface design-choices.

The desired organization and structuring of recommendations and the number of items shown affects the appropriateness of both modeling and evaluation techniques. Commonly, the user-experience and user-satisfaction are leading in determining the best method of presenting recommendations to users. An important design choice that should be considered is the trade-off between information completeness and information overload [95; 100]. Showing too few items may have users wandering whether they are missing out on items that are not shown. On the other hand, showing too much items can be overwhelming. The number of items recommended at the time, may favour certain recommendation techniques. Tailored models and algorithms have been developed to facilitate towards certain options, e.g. sequential models tailored for recommending a sequence of items [101] or models that explicitly consider the position of items in a list [102]. In addition, it should be noted that certain evaluation metrics, as discussed in Section 3.2, may only be applicable to a list of items, rather than a single recommendation.

News content may be presented in a multitude of modalities, such as text, video, audio or image, which makes it especially challenging to model and compare items [103]. In addition, news readers may wish to read or consume news in multiple languages, which poses similar challenges. To be able to recommend relevant items from a range of modalities or in different languages, it is required to make some sort of comparison between news items in different formats or languages. Many embedding techniques are tailored or developed for a single format (e.g. images or text) and many natural language processing methods are monolingual [104]. This makes it challenging to develop a modeling framework in which news items are represented uniformly. News recommender systems that are effectively capable at recommending multi-modal and multi-lingual items may be perceived as more useful by users, even if they suffer a drop in accuracy performance.

4.2 Organizational Context

Although recommender systems are designed to assist users, they are often implemented to serve certain business or organizational objective as well [95]. Hence, recommender systems may have various (and sometimes competing) objectives to serve the needs and desires of multiple stakeholders. In addition, the objectives and goals may not be uniform across a single group of stakeholders, e.g. users may have different objectives depending on their personal preferences. Jannach [95] and Jannach and Adomavicius [105] classify two (groups of) stakeholders: users and the system provider. In this section, we explore the goals and objectives of this latter stakeholder, the system provider or organization, as this reveals insights about the organizational context in which news recommender systems reside.

Organizational Goals The organization that designs and implements a news recommender system, the system provider, usually does so with certain business objectives in mind, such as revenue, profit, user-retention, or user-satisfaction. Some of these objectives, such as user-satisfaction, serve both the users and the organization, but there may also be competing interests. Conflicting objectives essentially arise as a result of the business model the system provider operates under. In the domain of (news) content recommendations, the prevailing business model relies on ad-revenue. Therefore, the system provider may have an intrinsic motivation or ulterior objective to recommend items that maximize ad-revenue, instead of user-satisfaction. In that case, the system provider should find a balance between the competing objectives of user-satisfaction and ad-revenue, as leaning too far to either side may be sub-optimal. Jannach [95] and Jannach and Adomavicius [105] provide a detailed discussion of the possibly competing organizational objectives that may arise for recommender systems in general.

4.3 Societal Context

Content recommender systems are able to provide highly personalized experiences, but also face substantial criticism in society. Critics argue that by continuously showing users content that aligns with their personal interests, platforms may limit exposure to diverse perspective and reinforce existing beliefs, which may lead to increased polarization in society. Furthermore, there are concerns about the lack of transparency and algorithmic accountability of the models used. These algorithms often operate as black boxes, with limited transparency regarding their inner workings, which raises concerns about potential biases and unintended consequences. In addition, privacy risks arise because recommender systems rely on collecting vast amounts of personal data from users to be able to infer users' preferences. Especially news recommender systems may be at risk of raising social and ethical concerns, due to the nature of news and the fundamental role news plays in any democratic society. Therefore, this section explores the societal and ethical context of news recommender systems, such that researchers and practitioners can mitigate social and ethical concerns.

4.3.1 Societal Role

Helberger argues that media, and consequently news recommender systems, have two important roles to fulfill in any democratic society [106]. The first objective is to inform citizens, to enable citizens to make an informed political choice, and to hold democratically elected representatives accountable. The second objective is to facilitate a public forum where different ideas and opinions can be articulated, encountered and debated. The work of Helberger provides a discussion of the relative weight different theories of democracy attach to these two roles. Here, we will refrain from discussing democratic theories at length, but it is worth noting that the role a news recommender system plays in a democratic society depends on the democratic theory one adheres to. For example, if the system design strives for a recommender system conform a liberal model of democracy, the preference should be given to self-selected recommendations (users determine the selection criteria for recommendations based on their preferences). On the contrary, conform a participatory or deliberative model of democracy, it may be more desirable to design a recommender system which occasionally nudges users towards more diverse viewpoints or information they "ought to read"[106].

4.3.2 Ethics

Recently, some academics have raised ethical challenges or concerns that should be understood and addressed during the design and deployment of recommender systems [107]. Milano et al. [107] formulate six areas of concern posed in the literature, namely "inappropriate content, privacy, autonomy and personal identity, opacity, fairness, and social effects". According to Milano et al. [107] a feature of a recommender system can be categorised as an ethical issue if "it negatively impacts the utility of some of its stakeholders or, instead, constitutes a rights violation, which is not necessarily measures in terms of utility". Below I will briefly summarize the six areas of concern as categorized by Milano et al. [107].

- *Inappropriate content:* Inappropriate content harms the user-experience. Mitigating the negative impact of inappropriate content (or items) in recommender systems is, thus, mainly an issue of quality control.

- *Privacy:* User privacy is inherently challenging for recommender systems, as most systems require user data to effectively model users' preferences. Privacy risks do not only arise because the system models users' preferences from the data, but also simply because such data is collected and stored. If data is collected or shared without explicit consent or data is leaked to malicious external agents, the users' privacy is at stake.

- *Autonomy and personal identity:* There may be a risk for users that recommender systems make certain recommendations to nudge users in a certain direction, by providing "addictive" content or by limiting the range of options.

- *Opacity:* In theory, transparent algorithms or explainable recommendations could mitigate the risks posed on the autonomy of users. Currently, however, transparency and explainability are not yet sufficiently achievable and still pose a challenge to academics.

- *Fairness:* There is an ongoing debate among academics about what fair recommendations should entail, albeit inconclusive, due to multiple notions of fairness, which are not all mutually compatible [108]. An example of two mutually incompatible notions of fairness that may arise in recommender systems is the conflict between equality of opportunity and equality of outcome [109]. That is, it may be debated whether it would be more fair to recommend underrepresented news-articles or minority viewpoints more often, such that they are recommended as often as any other news (equality of outcome), or whether it would be more fair to give each article, author or publisher the same opportunity to be recommended (equality of opportunity).

- *Social effects:* The ethical challenge faced by recommender systems, which is probably brought up most often both in- and outside of academia, is the impact recommender systems may have on society. Especially news recommender systems and social media have a high risk of creating self-reinforcing biases and creating “echo-chambers” or “filter bubbles”. Researchers have pointed out that these effect may be damaging to the normal functioning of public debate and to democratic institutions [90; 110; 111; 112; 113].

Researchers have proposed a variety of methods and practices, which may be adopted in the design process of news recommender systems, to mitigate some of the ethical challenges raised by Milano et al. [107]. For example, Wang et al. [114] developed a multi-task framework which is able to generate textual explanations for recommendations, which may improve transparency. Mulder et al. [115] propose a framework to increase viewpoint diversity in news recommendations, which may mitigate adverse social effects. Furthermore, Qi et al. [116] propose a novel news recommendation framework, which allows user behavior data to remain stored on the users’ devices, thus alleviating privacy concerns. In addition, both Helberger [106] and Heitz et al. [117] have argued that news recommender systems may also have positive effects on a democratic society.

5

Case Study

This section presents a case study in which we carry out a practical investigation of human-centered deep learning for news recommender systems. This case study builds upon the deep learning models introduced in Section 2 and the insights and evaluation metrics from Section 3. The case study is conducted to examine the state-of-the-art in deep learning methods for news recommender systems and to review current capabilities of moving towards a human-centered research approach. By implementing the deep learning models introduced in Section 2, this work provides both a practical and theoretical starting point for researching state-of-the-art deep learning models for news recommender systems (research-question SQ1). In addition, the deep learning models are implemented in an end-to-end multi-stage recommender system in this case study. Therefore, our case study provides practical insights in the interplay between candidate retrieval methods and ranking methods, which to the best of my knowledge has not been done prior in the domain of news recommendations. In doing so, this work aims to take a step towards a broader perspective of deep learning models for news recommender systems to avoid the framing trap introduced in Section 1. The experiments are conducted on the Microsoft News Dataset (MIND) [5], which serves as a benchmark dataset for news recommender systems. Previous academic research has focused mainly on developing novel deep learning methods for achieving improved accuracy performance. This work, instead, aims at exploring the possibility of evaluation deep learning models following a human-centered approach. In doing so, we aim to provide practical insights to support our understanding of what users want and how this can be quantified. That is, we aim to provide practical insights for our second research-question SQ2.

The remainder of this case study is structured to follow the theory provided in Section 2 and Section 3. Section 5.1 introduces several publicly available datasets and substantiates the decision for using the MIND dataset for this case study. Section 5.1.1 describes the MIND dataset and reports general statistics to establish a thorough understanding of the data. Section 5.2 describes the models used for the experiments and the methodology. Section 5.3 evaluates the recommendation models on accuracy performance. Section 5.4 evaluates the diversity of the recommendations, using two methods. In Section 5.4.1 diversity is measured using a domain-independent intra-list similarity metric, using the CosSim metric. In Section 5.4.2 diversity is determined using the intra-list similarity metric based on the latent representations of news items. Section 5.5 evaluates the novelty of recommendations, which is first done on an item-level in Section 5.5.1. In Section 5.5.2 this work proposed an adapted version of the novelty metric of [79] and [78] which operates on a category-level. Finally, the conclusions based on the experiments are formulated in Section 5.6.

Objective The objective of this case study is to provide a starting point and practical recommendations for researchers and practitioners interested in adopting a human-centered approach to deep learning models for news recommendations. This case study mainly addresses two aspects, which we consider essential to transition to a human-centered approach. First, this case study aims to take a step in overcoming the formalism trap defined in Section 1, by considering qualities of recommendations other than accuracy, such as diversity and novelty. Second, this case study serves as a testimony for adopting a broad perspective in academic research, in order to avoid the framing trap (see Section 1). This work explicitly examines an end-to-end multi-stage recommender system in an attempt to reveal insights for further research.

5.1 Datasets

There are several publicly available datasets for news recommender research, such as Plista [118], Adressa [119], Yahoo [120], and MIND [5]. Both Raza and Ding [19] and Karimi et al. [30] provide an overview of these and other news recommendation datasets. The Plista dataset consists of data gathered from 13 German news portals, the Adressa dataset consists of users' reading time of Norwegian news articles and the Yahoo and MIND dataset are both gathered from English news articles. This research is conducted for a Dutch organization, therefore, it seems counterproductive to select a dataset in a different language. To the best of my knowledge, there is however no publicly available dataset which contains user logs for Dutch news articles. Hence, an English dataset is preferred, because (1) there are more Natural Language Processing (NLP) techniques available for the English language and (2) English is understood by a broader audience.

Both the Yahoo dataset and the MIND dataset serve as a good benchmark dataset for researchers. There are, however, two key differences between the two that should be considered. First, the Yahoo dataset consists of explicit user ratings, which may make it the preferred option for collaborative filtering (CF) research [19]. The MIND dataset, on the other hand, logs user feedback in the form of click and non-clicked events. Secondly, in the Yahoo dataset the actual content of the news items is unavailable, whereas the MIND dataset contains rich textual content including the title, abstract, body, and category [19]. In the intended application setting for which this research is conducted, the user feedback is assumed to be collected as implicit feedback, that is in the form of click and non-clicked events (or reading time). Also, it is assumed that there will be access to similar rich textual features as in the MIND dataset. In addition, the MIND dataset is easily accessible online, whereas the Yahoo dataset are only made available upon request. Therefore, the MIND dataset will be used for this research.

5.1.1 Microsoft News Dataset (MIND)

The MIND dataset contains user logs collected from the Microsoft News website from October 12 to November 22, 2019 [5]. The dataset contains anonymized behaviour logs of 1 million randomly sampled users who had at least 5 clicks during this period. Microsoft provides two versions of the dataset, a large version which includes all impression logs generated by the 1 million users, and a small version created by randomly sampling 50.000 users and their behaviour logs. In addition, both versions are already split in a training and test set to promote accurate comparison among publications. For the large version of the dataset, the first four weeks of the collection period are used to construct users' click history, the fifth week is used for training and the sixth week for testing. In addition, the samples in the last day of the fifth week are used as a validation set. The test set, however, does not contain labels, as this test set has been used for the MIND News Recommendation Competition held from July to September 2020, as part of the 1st International Workshop on News Recommendation and Intelligence, held on April 14, co-located with The Web Conference 2021.[1] Therefore, predictions made on the test set can only be evaluated

[1] At the time of writing, information about the workshop can be found at `https://msnews.github.io/workshop.html` and the winning submissions for the competition, including technical reports, can be found at `https://msnews.github.io/competition.html`.

File	Name	Description
1	behaviors.tsv	Contains the click histories and impression logs of users.
2	news.tsv	Contains the information of the news articles.
3	entity_embedding.vec	Contains pre-constructed 100-dimensional embeddings for the entities in each news article (learned from WikiData by TransE method).
4	relation_embedding.vec	Contains pre-constructed 100-dimensional embeddings for the relations between entities in each news article (learned from WikiData by TransE method).

Table 5.1: Description of MIND dataset files.

by submitting results to the MIND competition hosted on CodaLab[1], which limits the evaluation metrics to AUC, MRR, nDCG5, and nDCG10. Hence, the validation set is used in this work for evaluating the models.

Both the training and validation set contain four different files, see Table 5.1. Since the data in production does not contain information about entities or relations between entities, unless extracted using NLP methods, the entities and relations and corresponding embeddings will be disregarded. Disregarding the entities and relations may make it harder to make accurate predictions. Nevertheless, there are numerous publications on the MIND dataset that do not make use of the entities and relations, such as [20; 21; 23; 121]. Therefore, we should still be able to compare results to other publications, even without taking the entities and relations into account.

The *behaviors.tsv* file contains the impression logs and users' click histories. It has five columns:

1. *Impression ID:* ID of an impression log.
2. *User ID:* Anonymized ID of the user.
3. *Time*: Impression time in the format "MM/DD/YYYY HH:MM:SS AM/PM".
4. *History:* List of news articles (News IDs) the user has previously clicked on.
5. *Impressions:* List of news articles (News IDs) displayed to the user in this session, including user's click behavior (1 for click and 0 for non-click). The order of news articles in an impression have been shuffled.

The *news.tsv* file contains all features related to the news articles, which are:

[1] Currently, the MIND competition is publicly available for anyone to participate in and being hosted on CodaLab at `https://codalab.lisn.upsaclay.fr/competitions/420`.

	Training set	**Validation set**	**Test set**
# Impression samples	2.232.748	376.471	2.370.727
# Unique users	711.222	255.990	702.005
# News articles	101.527	72.023	120.959

Table 5.2: Dataset statistics

1. *News ID:* ID of the news article.
2. *Category:* One of twenty keywords describing the category of the news article.
3. *Subcategory:* Keyword further specifying the category of the news article.
4. *Title:* Full text of the title.
5. *Abstract:* Full text of the abstract.
6. *URL:* URL of the news article on the Microsoft News website.
7. *Title Entities:* Information about the entities in the title. *(disregarded)*
8. *Abstract Entities:* Information about the entities in the abstract. *(disregarded)*

According to the original paper accompanying the release of the MIND dataset, there should be 2.186.683 samples in the training set, 365.200 samples in the validation set, and 2.341.619 samples in the test set [5]. This deviates from the number of samples in the datasets used for this research[1], which are shown in Table 5.2. It is unclear why, as the datasets were downloaded from the official source. Abdulhussein and Obaid [122] also report a different number of samples in their work, namely 1.000.000 samples for the training set and 1.048.576 samples for the test set. Both Abdulhussein and Obaid [122] and Ruan [123] do report the same number of news articles in each dataset as mentioned in Table 5.2. Furthermore, the original paper mentions that the dataset is contructed by sampling data from 1 million users, but in the dataset used in this work, there are only 876.956 unique users in all three datasets combined. It might be the case that they did sample 1 million users and thereafter removed users with less than 5 news clicks during the sampling period, but this is unclear from the original paper [5], so one can only guess.

The MIND dataset serves as a benchmark dataset for ranking models. That is, for each user in the *behaviors.tsv* file, a set of candidate news articles (impressions) is provided, and the goal is to rank these news articles as appropriately as possible. As shown in Table 5.1.1, there are on average 37,40 news-articles to be ranked for each user. Essentially, the candidate retrieval stage, that is the stage in which a small set of candidate items is selected from the entire catalog, is already done. In the intended use case, such a small

[1] Datasets are downloaded from the official website(`https://msnews.github.io/`) on 11/05/2023.

News dataset	**Training set**	**Validation set**	**Test set**
# Categories	18	17	18
# Subcategories	285	269	290
Avg. title length (words)	10,68	10,74	10,71
Avg. abstract length (words)	36,44	35,44	36,50
Impression log dataset	**Training set**	**Validation set**	**Test set**
Avg. history length	33,00	32,65	41,61
Avg. # impressions	37,40	37,41	39,28
Avg. click-ratio	0,108	0,10	NA

Table 5.3: Feature statistics

subset of candidate items is not readily available for each user and has to be constructed using candidate retrieval methods. In this case, candidate retrieval methods would have to select a subset of candidate news articles from the *news.tsv* file, which contains 101.527 news articles. Since the candidate retrieval stage has already been performed, the number of candidate items that have to be compared is small. This allows us to evaluate more complex ranking models on the dataset.

Most of the time, users only click on one article from all the articles presented in one session. In fact, this is the case in 1.613.818 of the impressions, which is 72,28% of the impression samples. On average, users click on 10,8% of the articles shown to them in a single session. Each news article in the dataset has been clicked on at least once by a user. On average, articles are clicked on by 660 users. The number of clicks on a single article is, however, skewed to the right, due to some very popular articles which have been clicked on almost 5.000 times.

5.2 Methodology

In this section, we first introduce several two-tower deep learning models developed for news recommendations, which have been implemented for this work and describe the corresponding methodology in Section 5.2.1. In the second part, in Section 5.2.2, we describe how these deep learning models were implemented in an end-to-end multi-stage news recommender system.

5.2.1 Two-tower Deep Learning Models

The methods used for this work were all developed on the MIND dataset and each adopt a two-tower deep learning approach as discussed in Section 2.1. In this work, we turn our attention to four news recommendation methods: NRMS [21], LSTUR [121], NAML [20] and TANR [22]. These methods were selected for two reasons. First, implementations of these models can be found in the Microsoft Recommenders open source repository [124], which aids in the ability to replicate the experiments presented in this work. Second, these models were among some of the earlier deep learning models proposed for news recommendations in recent years and have since been referenced by numerous researchers. Below, we will briefly describe each method. For the full implementation details of each model readers may refer to the respective publications [20; 21; 22; 121] .

NRMS The NRMS model proposes a neural news recommendation method with multi-head self-attention, which learns news representations from the news titles and user representations from previously clicked news articles. Multi-head self-attention is adopted to model the interactions between words in the title, as well as the relations between previously clicked news articles.

LSTUR The LSTUR model proposes a neural news recommendation method with long- and short-term user interests. News representations are learned from their titles and topic categories. Long-term user interests are learned from the embeddings of users' IDs, while short-term user interest are learned from their recently clicked news articles via a GRU network.

NAML The NAML model proposes a neural news recommendation method with attentive multi-view learning to capture a unified news representation from the title, body and topic category of a news article. User representations are learned through an attention mechanism that selects informative news from the news articles users have clicked on in the past.

TANR The TANR model proposes a neural news recommendation method with topic-aware news representations. This method adopts a multi-task learning approach to rank the relevancy of news articles and to predict the topic of the article. News representations are learned from their titles via CNN networks and an attention mechanism is employed to select important words in the title. The user representations are learned using an attention network to select informative news articles from the articles users have clicked on in the past.

The implementation of the models used for this work is an adaptation of the work of Ning [125], which is more intuitive and flexible than the implementation of Microsoft [124]. The same experimental settings for training the models are used as in the original works. Only for the LSTUR and NAML model, the batch size during training is reduced from a batch size of 400 and 100, respectively, to a batch size of 64, in order not to put to much strain on computational resources. For a full description of the implementation details and parameter setting, readers may refer to Appendix Section 8.2.

The NRMS, LSTUR and TANR model were all trained on a machine with an NVIDIA T4 Tensor Core GPU. This work reports the following training times. Training the NRMS and TANR model both took about 2 hours and 30 minutes wall-clock time, while training the LSTUR model took 4 hours. The NAML model was trained on a machine with an NVIDIA V100 Tensor Core GPU, which took 3 hours. Training the NAML model on a T4 Tensor Core GPU, like the other models, is estimated to take about 12 hours.

5.2.2 Multi-stage Recommender System

Given the fully trained two-tower deep learning models described in the previous section, we can use these models to implement an end-to-end multi-stage news recommender system. To achieve such an end-to-end multi-stage recommendation system this work adopts the following methodology. The trained two-tower deep learning models each contain a trained module for encoding news articles as their latent representation. These trained news-encoders are used to construct a latent representation for each news article in the entire catalog, i.e. in the *news.tsv* file of the MIND dataset, which are subsequently stored in a vector-database implemented using Faiss [56]. To retrieve a relevant set of candidates for a certain user, the user is mapped to its latent representation using the trained user-encoder module of the deep learning models. Next, vector-based similarity-search is utilized to retrieve the latent representations of the 20 news items most similar to the latent representation of the user. These retrieved news items, subsequently, serve as the set of candidates that is fed to one of the two-tower deep learning models in the previous section. These two-tower models rank the set of candidates for the given user based on the relevance score for each candidate.

	AUC	**MRR**	**nDCG5**	**nDCG10**
Random baseline	0,4892	0,2198	0,1688	0,1922
NRMS	0,6619	0,3190	0,3102	0,3401
LSTUR	0,6649	0,3161	0,3090	0,3367
NAML	**0,6778**	**0,3268**	**0,3203**	**0,3480**
TANR	0,6097	0,2854	0,2665	0,2867

Table 5.4: Accuracy performance ranking models

5.3 Accuracy Evaluation

First, the performance of the models introduced in Section 5.2 is evaluated based on accuracy only. Similar to previous works, the accuracy of the models is evaluated using the AUC, MRR, nDCG5 and nDCG10 metrics (see Eq. 8.3, Eq. 8.4, and Eq. 8.6). The results are shown in Table 5.4. In addition to the AUC, MRR and nDCG the precision and recall of each model is also evaluated at different cut-off values k (Eq. 8.10 and Eq. 8.11). Figure 5.1 and Figure 5.2 show the precision and recall of all models evaluated at different values of k, where k represents the top-k highest ranked items in a set of recommendations.

The NAML model has the best performance across all accuracy metrics, closely followed by the NRMS and LSTUR model. The accuracy performances reported in this case study is similar to the performance reported by Wu et al. [5], however, they report the NRMS model to outperform the LSTUR and NAML model. The performance reported in [21], [121], and [20] is slightly lower than the performance reported here. The observed differences in performance are likely due to deviations in the test sets used, but may also be caused by slight deviations in the implementation of the models.

5.4 Diversity Evaluation

The diversity of recommendations is an important aspect in the user-satisfaction [2; 3; 4]. Here we evaluate the diversity of recommendations made by state-of-the-art deep neural ranking models using various methods proposed in academic literature. The intra-list similarity (ILS) metric (Eq. 3.1 or Eq. 3.2) seems to be the most common approach for evaluating the diversity of recommendations. Here we adopt two methods for determining the ILS. In Section 5.4.1 the diversity of recommendations is evaluated using the ILS based on the CosSim metric (Eq. 3.4). In Section 5.4.2 diversity is evaluating using the ILS based on the latent representations of news items. We limit our discussion to the LSTUR and

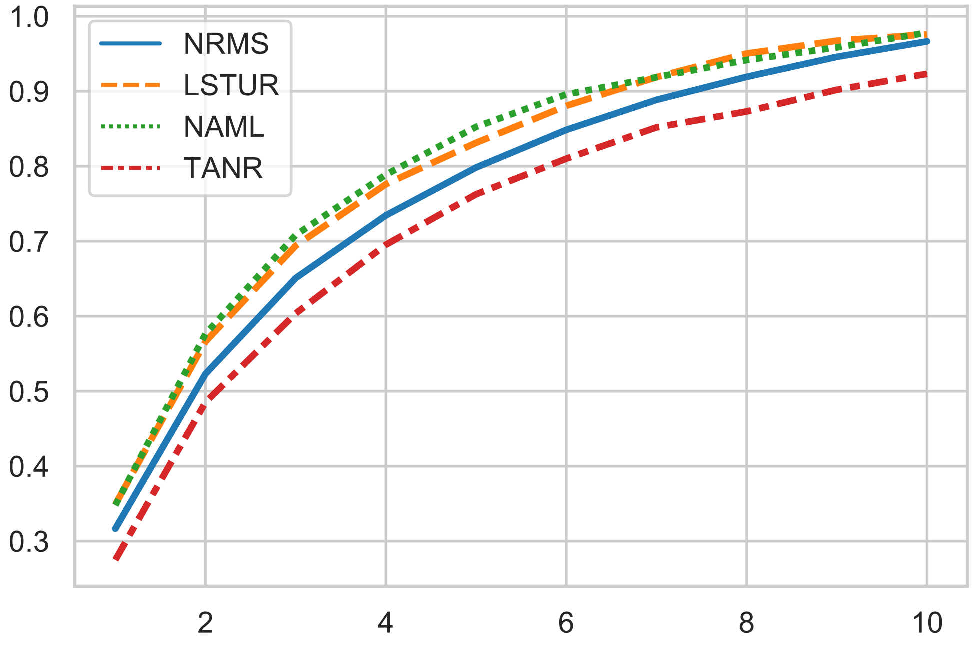


Figure 5.1: Precision at k

NAML model, because these models show very similar performance in our experiments, but are very different in their modeling approach.

5.4.1 Domain-independent intra-list similarity

Determining the ILS metric using the CosSim metric seems a favourable approach from a theoretical standpoint, but in this case study we encountered several practical complications. The CosSim metric determines the similarity between two items based on the fraction of users that like both items, instead of a latent representation of the items, which makes it domain-independent in theory. However, the experiments conducted in this work raise two practical concerns regarding the suitability this particular method in the context of news recommendations. First, this method requires tremendous computational effort. Second, the experiments show that it may not be applicable in recommendation scenarios where the catalog changes frequently and users only interact with a small fraction of items, as is the case in news recommendations.

The computational effort required to compute the CosSim metric severely limits the applicability of the method in this case study. Determining the number of users that

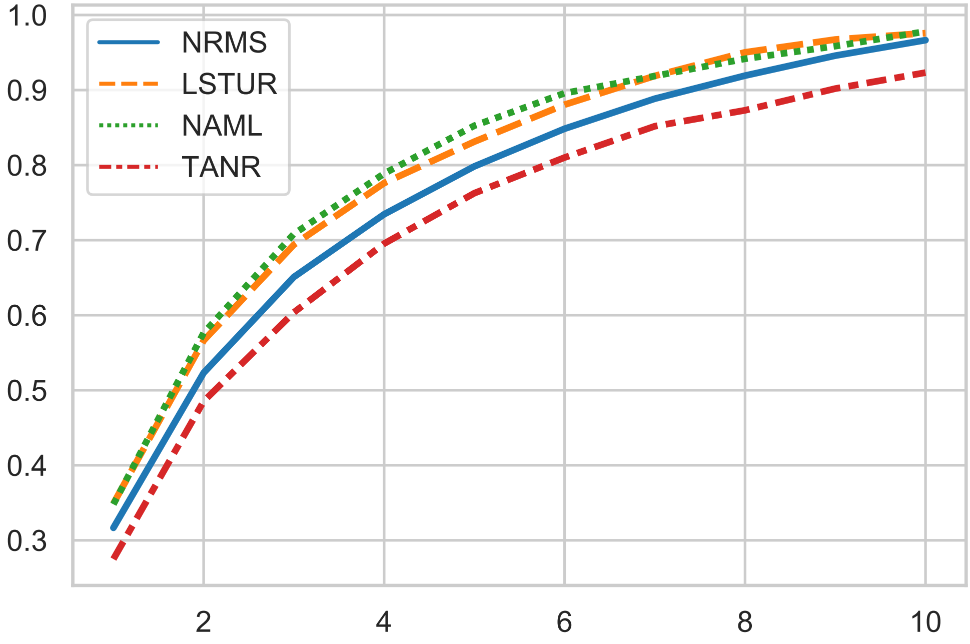


Figure 5.2: Recall at k

like both item i and item j requires significant computational effort. To determine the numerator of the CosSim metric (i.e the number of users who like both item i and item j), we have to compute all pairs of items $[i, j]_{i \neq j} \in H_u \forall u \in U$. With 711.222 users, this results in about 375.525.216 item-item pairs in the training set, estimated by:

$$\sum_{u=1}^{|U|} \frac{|H_u|(|H_u| - 1)}{2},$$

and the average number of items in the history of a user (33,00). Therefore, this method requires extensive computational resources and memory, which were not available for this research. To be able to calculate the CosSim, while staying within a reasonable computational budget, sampling was attempted in this study. We sampled 10.000 users from the training set and 10.000 impressions from the recommendations to evaluate the intra-list similarity. The results of repeating this experiment 10 times are shown in Figure 5.3.

The results reported in Figure 5.3 highlight the second practical complication encountered in this work. The reported ILS scores are extremely low, which may be attributed to the domain characteristics of news recommendations. The number of news items liked

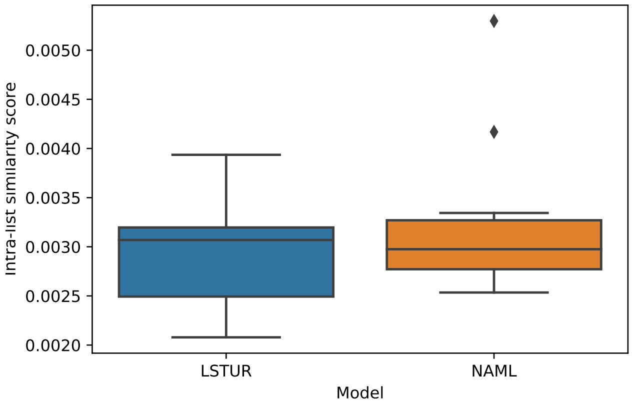


Figure 5.3: Domain-independent intra-list similarity of LSTUR and NAML model at $k = 5$.

(read) by users (33,00) is considerably smaller than the number of items in the catalog (101.527), therefore, the chance that a user likes (has read) two news articles that appear together in a list of recommendations is very slim and often zero. In fact, for both the LSTUR and NAML model the number impressions for which the ILS is zero is over 99%. Thus, we conclude that it is questionable whether the CosSim metric is adequately able to capture similarity (or diversity) as perceived by users in this domain. Due to the substantial size of the catalog and the high rate of changes in the catalog, as well as the fact that the number of items liked by users is relatively small, the CosSim metric seems to perform poorly.

Given the fact that, in this application, the likelihood of users liking two items occurring together in a list of recommendations is extremely low, it can be theorized that several other evaluation metrics discussed in Section 3.2 will encounted similar limitations as the CosSim metric. Having established that the CosSim metric in Eq. 3.4 is poorly applicable in this case, determining the serendipity following Eq. 3.16 will likely also be troublesome. Similarly, calculating the unexpectedness of recommendations using Eq. 3.20 or Eq. 3.21 based on Eq. 3.19 will suffer from the same limitation.

5.4.2 Intra-list similarity using latent representations

Several researchers have proposed to use the distance between latent representations of recommended items as a metric for how similar or dissimilar items are [11; 71; 126; 127]. A major advantage of using latent representations to determine the similarity between two items is that the latent representations are already computed. The drawback of using latent representations, however, is that constructing appropriate latent representations is highly context-dependent, which limits the ability to compare recommendation methods across domains. In addition, there is often a lack of validation whether the distance between latent representations coincides with users' perception of diversity [70]. Comparing the diversity of recommendation methods is further hindered by the fact that academic publications often omit a detailed description of the latent representations used for evaluation.

Based on the experiments conducted for this research, we argue that using the latent representations of items may be an appropriate method, but drawing conclusive insights poses its challenges. First of all, the experiments highlight the importance of using the same method for constructing latent representations to be able to compare results. In addition, the experiments demonstrate the need for validating whether latent representations coincide with users' perception of diversity through user-studies. Furthermore, the results presented here hint at a possible increase in diversity that may be obtained from combining several retrieval methods in multi-stage recommender systems.

The ILS-score of the recommendations of the LSTUR and NAML model are evaluated over the top five recommended items for each impression (cut-off $k = 5$) after sampling 10.000 impressions. The cosine-similarity is calculated as the normalized dot-product between the two latent representations of the items. The latent representations of the news items are created by the trained news-encoder modules of the LSTUR and NAML model. The results of sampling and evaluating the recommendations 10 times are shown in Table 5.5. If we compare the LSTUR and NAML model using the ILS-score that is calculated using the same news-encoder, the results are very similar. In both cases, the ILS-score lies closely together and the ILS-score of the NAML model is only marginally higher. The results presented in Table 5.5, however, clearly indicate the importance of using the same latent representation to evaluate the diversity of different models. If we would use the latent representation created by the LSTUR news-encoder to evaluate the LSTUR model, and the latent representation created by the NAML news-encoder to evaluate the NAML model, we might falsely come to the conclusion that the LSTUR model provides more diverse recommendations, based on the ILS-scores of 0,2455 and 0,4603, respectively (lower ILS-score indicated higher diversity).

The news items in the top 5 recommendations used for evaluating the diversity of the LSTUR and NAML model in Table 5.5 are likely very similar in both cases, since the set of candidates is the same for both models and both models perform similarly on precision and recall. Therefore, it seems reasonable that the ILS-score is similar for both models, if the same latent representation is used for evaluation. However, if the set of

Model	Encoder	ILS
LSTUR	ILS-LSTUR	0,2455
	ILS-NAML	0,4461
NAML	ILS-LSTUR	0,2498
	ILS-NAML	0,4603

Table 5.5: Intra-list similarity score of LSTUR and NAML model at $k = 5$ depending on which news-encoder (ILS-LSTUR or ILS-NAML) is used to determine similarity between latent representations of items.

candidates is changed for each model we obtain very different results. Table 5.6 shows the ILS-score for the top 5 recommendations for the LSTUR model and NAML model after selecting a set of 20 candidates using similarity-search candidate retrieval. The ILS-score is, again, evaluating using either the latent representations created by the LSTUR news-encoder (ILS-LSTUR) or the latent representations created by the NAML news-encoder (ILS-NAML). By adopting similarity-search based candidate retrieval it is expected that the set of candidates, and thus the top 5 recommendations, are much more similar. This is indeed the case for the LSTUR model, where the ILS-score is high, regardless of the latent representation used for evaluation. For the NAML model, however, the different ILS-metrics produce significantly different results.

The experiments show significantly different ILS-scores for the NAML model, based on which news-encoder is used to construct the latent representations. Thus, the recommendations made by the NAML model are highly similar according to the latent representation of the NAML news-encoder, but much more diverse according to the latent representation of the LSTUR news-encoder. To further illustrate this observation, two examples of recommendations that result in different ILS-scores are shown in Table 8.1 in Appendix 8.3. In both cases, the similarity between recommendations is high according to ILS-NAML, but low according to ILS-LSTUR (Example 1: 0,8863 and 0,3599, respectively) (Example 2: 0,7832 and 0,3439, respectively). The difference may likely be attributed to the fact that the LSTUR news-encoder and NAML news-encoder adopt different paradigms for modeling news items. Based on the examples shown in the Table 8.1 it seems as if the LSTUR news-encoder assigns more weight to the category and sub-category, while the NAML model assigns more importance to the title and abstract of a news item.

To conclude, like Jesse et al. [70] our experiment shows the importance of verifying the validity of diversity metrics through user-studies, but also hint at the possibility of improving diversity without loss of accuracy. The recommendations in example 1 shown

Multi-stage system	**Encoder**	**ILS**
LSTUR	ILS-LSTUR	0,6778
	ILS-NAML	0,7398
NAML	ILS-LSTUR	**0,3745**
	ILS-NAML	0,6919

Table 5.6: Intra-list similarity score of multi-stage recommender system based on LSTUR or NAML model at $k = 5$ depending on which news-encoder (ILS-LSTUR or ILS-NAML) is used to determine similarity between latent representations of items.

in Table 8.1 may be perceived as highly similar by users, due to the fact that all articles contain some list of facts about a certain topic. However, the same recommendations may also be perceived as highly diverse, because each article covers widely different topics, e.g. animals, Halloween and felons' rights. Similarly, in example 2, each news article seems to be food and drinks related, which may be perceived as similar articles. Nonetheless, one article discusses the share-price of a large restaurant chain, while another is about a pie-eating contest, which users' may perceive as very different topics. Determining which method of evaluating diversity coincides with users' perception of diversity, thus, requires user-studies. At the same time, the LSTUR and NAML model achieve comparable accuracy performance, even though news items are mapped to very different latent representations. Thus, it seems possible to combine both models to enhance diversity without the loss of performance.

5.5 Novelty Evaluation

The notion of novelty refers to a recommender system's capacity to present users with items they haven't encountered before [4]. Previous research indicates that including more novel recommendations can have a positive impact on user satisfaction [8; 12]. Here, we first evaluate the novelty of individual items recommended to users in Section 5.5.1. Next, we evaluate the novelty of the categorical labels of news items recommended in Section 5.5.2. It is theorized that evaluating the recency of items is particularly important in the news domain to determine the freshness of recommendations. The most intuitive and straightforward method to evaluate the recency of recommendations is based on the time between the moment an item first enters the system and the moment the item is being recommended (Eq. 3.9). The MIND dataset, however, does not contain the data required to evaluate recency. Thus, this section limits its evaluation to the novelty of items and categories as calculated by the novelty metric of [78] and [79] based on the Shannon entropy.

5.5.1 Novelty of news items

To evaluate the degree of novelty of the recommendations, we adopt the novelty metric of Eq. 3.6 or Eq. 3.8), which are synonymous. To determine the novelty using this metric, the novelty of each recommended item needs to be determined using:

$$novelty(i) = -\log_2 p(i),$$

where $p(i)$ is the probability that item i is chosen by a random user, which is assumed to reflect the popularity of an item i. Intuitively, it would make sense to estimate $p(i)$ based on how often users in the training set have interacted with the item i, that is:

$$p(i) = \frac{\sum_{u=1}^{|U|} \mathbb{1}_i(H_u)}{\sum_{u=1}^{|U|} |H_u|}.$$

However, in this case this method poses its challenges, due to the unique characteristics of the news domain. Because the lifespan of news items is relatively short, many recommended items have never been seen before by users in the training set (81,39% of recommendations). These items are consequently neglected in evaluating the novelty metric, which severely limits the applicability of the novelty metric.

Here, we propose to estimate the popularity $p(i)$ of an item based on the items users have chosen in the lists of recommendations:

$$p(i) = \frac{\sum_{u=1}^{|U|} \mathbb{1}_i(R_u)}{\sum_{u=1}^{|U|} |R_u|}.$$

In that case, the time-frame over which $p(i)$ is estimated, coincides with the time-frame of recommendations that are evaluated, such that $p(i)$ can be estimated for all items i. Next, the novelty of items over a list of recommendations is calculated at different cut-off values k using Eq. 3.8. The results are shown in Figure 5.4. As expected, the novelty score is almost identical for the LSTUR model and the NAML model if they are presented with an identical set of candidates. However, the difference becomes more apparent if the candidates are selected using similarity-search. In that case, the LSTUR model seems to be able to provide more novel recommendations than the NAML model.

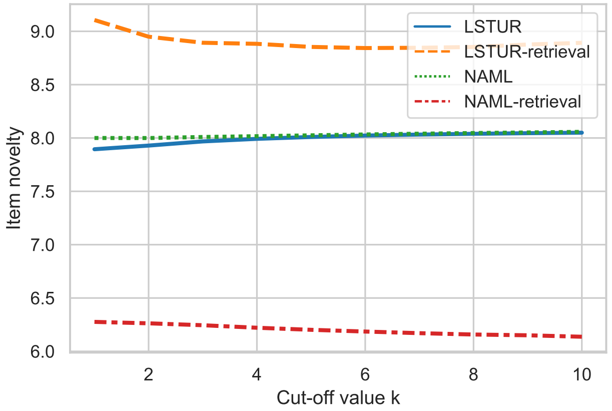


Figure 5.4: Item novelty of LSTUR and NAML model at different cut-off values k.

5.5.2 Novelty of news categories

Determining the novelty of items as done in the previous section seems to be highly sensitive to the lifespan of news items, which hinders the applicability and interpretability of results. The time-frame which is selected to determine the popularity $p(i)$ of an item may severely impact the obtained novelty-score. Therefore, this work proposes an adaptation of the novelty metric of Castells et al. [79] and Zhou et al, [78], which operates on a category-level, rather than an individual item-level. Instead of determining the popularity $p(i)$ of a single item, this work proposes to determine the popularity of the categorical label of a news item. Unlike individual news items, the categorical labels of news articles are static through time and each category will be read by numerous users. Therefore, evaluating the novelty of recommendations on a category-level will provide more stable results. Thus, we evaluate the categorical novelty of the LSTUR and NAML recommendations using a variation of Eq. 3.6 and Eq. 3.8, where $p(i)$ represents the probability that a news item with the categorical label of item i is chosen by a random user. Then, $p(i|R)$ denotes the probability that a user picks a news item with the categorical label of item i from the list of recommendations R. Since the categorical labels are static, $p(i)$ can be accurately estimated using the training set only.

The results of evaluating the categorical novelty of the LSTUR and NAML model are shown in Figure 5.5. The LSTUR and NAML line show the categorical novelty for the recommendations on the MIND dataset using the retrieved candidates as they are in the dataset. In that case, the categorical novelty of the recommendations is almost identical, which is expected, since the LSTUR and NAML model achieve similar performance, i.e.

the models will likely make very similar recommendations given the same candidates. Instead of using the same candidates for both models, we adopt vector-based similarity-search using Faiss [56] to retrieve a set of 20 candidates for each impression. In both cases, the vector database is constructed using the latent representations of the news-items as created by the news-encoder module of the respective model. Using such a similarity-based retrieval method, the set of candidates is expected to be highly similar. The results shown in Figure 5.5, however, indicate that the LSTUR-retrieval model is able to provide more novel (categorical) recommendations than the NAML-retrieval model. Thus, although the NAML model has a slightly higher accuracy performance than the LSTUR model, the LSTUR model may provide a better user-experience, since the categorical novelty of recommendations is higher.

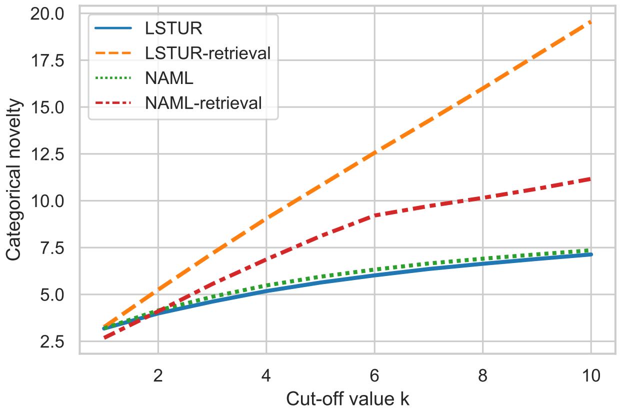


Figure 5.5: Categorical novelty of LSTUR and NAML model at different cut-off values k.

5.6 Conclusion

This case study examines current state-of-the-art deep neural ranking models for news recommendations. Four different deep neural ranking models (NRMS, LSTUR, NAML, TANR model), each with a different modeling paradigm are replicated from previous research. Our experiments report fairly similar accuracy performance, although there are some minor deviations. Next, the recommendations of the LSTUR and NAML model are analyzed in-depth for diversity and novelty. To evaluate the diversity of recommendations, two approaches using the intra-list similarity were used. One based on the CosSim metric of Zhang et al. [4] and one based on the latent representations of news items. In addition, the impact of the retrieved set of candidates on diversity is explored. This is done by using the news-encoder modules of the LSTUR and NAML model to construct a vector database (Faiss [56]) which supports vector similarity-search methods for retrieval. In the second part of the case study, the novelty of recommendations is evaluated on an item-level and category-level, using Shannon entropy.

Based on the experiments conducted here, the following conclusions can be derived. First, CosSim metric of Zhang et al. [4] seems to be largely inapplicable for determining the intra-list similarity in recommendation domains where the catalog is large and the number of item-interactions small, and where each item has a relatively short lifespan, such as the news domain. Second, evaluating diversity using the intra-list similarity based on latent representations of news items may be an appropriate method, but different latent representations may score the diversity of items in different ways. Hence, further research and user-studies are required to validate whether latent representations coincide with users' perception of diversity. At the same time, this observation hints at the possibility of increasing diversity without loss of performance, by combining different retrieval methods based on different latent representations. In addition, based on the experiments conducted here, the importance of using the same latent representations to evaluate the diversity of recommendations across multiple models is demonstrated. Furthermore, this work advocates for indicating the specific methods used to construct latent representations for evaluating diversity in academic works, as this will support reproducibility. Finally, this work demonstrates that evaluating the novelty of recommendations on an item-level may have its limitations in the news domain. Instead, this work proposes a novel category-level novelty metric, which may be more suitable in evaluating the novelty of news recommendations.

6

Discussion

This thesis aims to provide an alternative perspective on the way academics and practitioners think about researching and developing deep learning models for news recommender systems. This work advocates for a human-centered approach to deep learning research, which is aimed at providing satisfying and useful recommendations, rather than merely accurate recommendations. In addition, this work advocates for taking different perspectives of news recommender systems and advocates for placing these systems in context to overcome the framing trap and discover new avenues for interdisciplinary research. This work attempts to provide a starting point for interdisciplinary research, by discussing the different contexts in which news recommender systems resides. There is, however, a tremendous amount of literature available within the field of expertise of these contexts, which cannot all be condensed in this work. In addition, the discussion of modeling techniques for recommending news items is limited to deep neural ranking methods and vector-based similarity-search retrieval methods. These techniques seem to be the predominant methods for news recommender systems, both in real-world applications and academic research. Nonetheless, there are countless other techniques that may be adopted for recommending news items.

The majority of recent publications on deep neural news recommendations follow the two-tower framework discussed in Section 2.1 [17]. These methods have been shown to provide remarkable performance, since they are much better equipped to capture users' preferences and the semantic meaning of news items than traditional methods. However, deep learning models also pose several challenges, due to their complexity and opaque nature, and require significant computational effort. Furthermore, it must be noted that the majority of academic research on deep learning models for news recommendations is limited to recommending English news articles. Recommending news articles written in any other language may pose unique challenges not encountered in English. In addition, there are some publications which evaluate their methods in another way than merely using accuracy metrics. Some publications evaluate the models in an online environment, for example through A/B-tests. It must be noted that such online evaluation methods may be more insightful than offline accuracy metrics. Nonetheless, such online tests too are limited in their capability to account for the full meaning of user-satisfaction.

Adopting a multi-stage recommender system framework as presented in Section 2.2, which consists of at least a candidate retrieval stage and a ranking stage yields significant improvements in the scalability and latency of news recommender systems. A common approach that works especially well in unison with deep neural ranking models are vector-based similarity-search methods. This work briefly discusses vector-based similarity-search methods for candidate retrieval, because these methods may play a significant role in the diversity, novelty, serendipity or unexpectedness of news recommendations. There is a plethora of publications available on information retrieval, which may prove to be highly relevant for developing appropriate candidate retrieval methods. There are, however, very limited number of publications which evaluate candidate retrieval methods specifically for the news domain or which evaluate multi-stage recommender systems in their entirety. Therefore, this work aims to support further research on the impact retrieval methods can have on the recommendations made by deep learning methods.

Section 3 attempt to unveil the objectives of users when using a news recommender system and the qualities of news items and recommender systems that users perceive as useful, in an attempt to move towards a human-centered research approach. In Section 3.2 several offline beyond-accuracy evaluation methods are discussed for evaluating the diversity, novelty, recency, serendipity, and unexpectedness of news recommendations, as each aspect may play a role in the user-experience. The review of beyond-accuracy aspects and evaluation methods is limited to the once presented in Section 3.2, because those methods were identified to be the most applicable in the context of news recommendations. There are, however, many more evaluation metrics proposed in academic literature and, perhaps, these aspects are still limited in capturing all intrinsic properties of news recommendations that contribute to user-satisfaction. For example, Vrijenhoek et al. [128] propose a set of metrics to capture different normative notions of diversity, instead of treating diversity as a single absolute. In addition, truly understanding what constitutes to more satisfying recommendations cannot be done through offline evaluation metrics only. Instead, there is a necessity for evaluating recommender systems in online environments and validating offline-metrics through user-studies.

In the latter part of this thesis, a case study on the MIND dataset [5] is presented, which aims to put provide practical recommendations and insights to complement the theory in Section 2 and Section 3. The case study demonstrates that current beyond-accuracy evaluation methods are highly domain-dependent and require careful implementation and evaluation to be able to draw meaningful conclusions. In addition, the experiments show that using latent representations of items to evaluate diversity can work well, but more research is required to validate whether latent representations coincide with users' perception of diversity. The case study is limited to four deep learning models, which each were developed on the MIND dataset, of which two are explored in-depth. There are, however, many deep learning approaches proposed in academic literature, some of which have been shown to provide superior performance on the MIND dataset compared to the models used in this case study [17]. In addition, the experiments conducted here indicate that minor changes in methodology may provide different results, therefore, it is assumed that replicating the results presented here in a different context or on a different dataset requires a careful approach.

7

Conclusion

This work has argued that the majority of deep learning research for news recommender systems is overly fixated on accuracy. Instead, this work advocates for an human-centered approach, which considers the user-experience and user-satisfaction as the ultimate objective. Therefore, this work aims to provide a starting point for researchers and practitioners to move towards a human-centered research- and design-philosophy for developing, implementing and evaluating deep learning models for news recommender systems, which we expressed in our main research question RQ. To realize such a shift towards a human-centered approach, this work has explicitly taken a broad, lateral approach to news recommender systems, in an attempt to overcome certain pitfalls that traditional, vertical research may fall into. The three pitfalls this work aims to guard deep learning researchers and practitioners for are listed once again below.

- **Formalism Trap:** Failure to account for the full meaning of social concepts [which] cannot be resolved through mathematical formalisms. (e.g. failure to account for the full meaning of user-satisfaction by merely relying on accuracy metrics.)
- **Framing Trap:** Failure to model the entire system over which a social criterion will be enforced.
- **Solutionism Trap:** Failure to recognize the possibility that the best solution to a problem may not involve technology.

In an attempt to transition towards a human-centered research approach, this work has undertaken both a theoretical and practical investigation. The theoretical investigation has been structured to take an increasingly broadened view on deep learning models and news recommender systems, which is shown in Figure 1.1 and reflected in the three subquestions each theoretical section aims to address. In Section 2 we examined the state-of-the-art in deep learning models for news recommender systems (SQ1). We showed that the majority of deep learning models developed for news recommendations follow a common two-tower framework, which consists of three modules: a user module, candidate item module and prediction module. The user module and candidate item module use NLP techniques and neural networks to create latent representations of users and news items, respectively. The prediction module predicts the relevance of candidate items for a specific user. In addition to providing a conceptual overview of two-tower deep learning models, this work has also presented a mathematical formulation. Next, a systemic overview of multi-stage recommender systems was presented, which forms the foundation of many leading recommender systems in real-world applications. As such, this work aims to establish a thorough understanding of deep learning models for news recommender systems, both in an academic and application setting.

After establishing the state-of-the-art in deep learning for news recommender systems, this work attempts to steer away from traditional research which is merely guided by accuracy as a measure of performance. In Section 3 this work aims to answer how we can form a better understanding of whether news recommender systems provide recommendations that users want (i.e. recommendations that are satisfying and/or useful) (SQ2). First, it is important to recognize that users may have various objectives or goals when using a news recommender system. Users goals may be formulated in terms of the items they wish to see, e.g "finding some good items", or in terms of personal motivations, such as the desire to be socially engaged. Furthermore, this work attempts to uncover the qualities of news items and news recommender systems that determine how users perceive the system. Users perception of a news recommender system may be dependent on certain qualities of individual news items, such as the credibility, likability, quality and representativeness of news items. News recommender systems, however, have very limited control over the content of news items. Therefore, it may be more interesting to examine the qualities of the news recommender system itself which contribute to a positive perception. Based on the works of Swearingen et al. [12] and Pu et al. [8] we can form a comprehensive understanding of the various aspects that determine whether users perceive a news recommender systems as satisfying or useful. The difficulty lies rather in how these qualities may be translated to beyond-accuracy metrics. This work provides several beyond-accuracy metrics to evaluate the diversity, novelty, recency, serendipity and unexpectedness of recommendations, which coincides with certain qualities identified by Swearingen et al. and Pu et al.. These beyond-accuracy evaluation metrics may provide a starting point for a more comprehensive set of evaluation metrics. Nonetheless, more research is required to evaluate whether beyond-accuracy metrics indeed coincide with users' perception of certain qualities.

This work has explicitly taken a broad perspective of news recommender systems, most prominently in Section 4. This section examines the context in which deep learning models operate to reveal opportunities for interdisciplinary research (SQ3). This work has argued that there are three main contexts in which news recommender systems reside, not taking into account the users, which are the application context, organizational context and societal context. By considering the application context in which a news recommender system operates, the performance of deep learning models can be tuned to match the functionality and interface of the application. Furthermore, this work has argued that in real-world applications news recommender systems have to balance user objectives with organizational goals. Hence, the organizational context has been investigated. Finally, the societal context may be especially important for news recommender systems, due to the precarious nature of news in society. If researchers and practitioners in the field of deep learning for news recommender systems are aware of the societal context, social and ethical concerns may be mitigated.

The latter part of this thesis present a case study, to provide a practical starting point for researchers and practitioners to move towards a human-centered approach to deep learning for news recommender systems. The experiments conducted demonstrate that current beyond-accuracy evaluation metrics are highly domain-dependent and require careful implementation. Using latent representations to evaluate the diversity of recommendations may be an appropriate method, but depends greatly on the method used to construct latent representations. Therefore, further research is required to validate whether latent representations accurately capture users' perception of diversity. In addition, two recommendations must be noted as a result of the experiments. First, the experiments demonstrate the importance of adopting a single method for constructing latent representations when evaluating multiple models against each other. Second, this work argues that the scientific community may benefit from adopting methods that promote reproducibility in evaluating diversity based on latent representations.

Several experiments were conducted to examine the novelty of news items in recommendations, which proved to be challenging. A common method for evaluating novelty, based on the entropy of news items, proved to be difficult to implement and evaluate, due to the unique characteristics of the news domain. News items typically have a very short lifespan and the rate of catalog changes is extremely high in the news domain. Therefore, novelty metrics based on the historical popularity of items may not suffice. Instead, this work proposes a category-level evaluation method, which may provide more stable results. Nonetheless, further research is required to be able to fully understand the role of novelty in news recommender systems.

Unlike previous research on deep neural recommender systems on the MIND dataset, this work has implemented an end-to-end multi-stage news recommender system. Vector-based similarity-search methods are adopted to perform candidate retrieval in conjunction with a ranking stage. The vector-database (Faiss [56]) is constructed using the latent representations of news items as constructed by the news-encoders of two-tower deep neural recommendation models. This work firmly holds the believe that advancements in news recommendations requires modeling and evaluating recommender systems as entire systems, instead of their individual components. The candidate retrieval stage may play a pivotal role in achieving more diverse and more satisfying recommendations for users with a diverse set of interests. The MIND dataset may not lend itself particularly well for evaluating both the candidate retrieval and ranking stage, since there is no straightforward way to determine the accuracy and recall of the candidate retrieval stage. Nonetheless, future research aimed at evaluating the interplay of candidate retrieval and ranking may prove to be particularly valuable in designing human-centered news recommender systems.

To conclude, deep learning methods already exhibit great capabilities for news recommender systems, however, their performance may be optimized by taking an alternative perspective at these models. By evaluating deep learning models on a wide set of accuracy and beyond-accuracy metrics, we may uncover qualities of news recommender systems that users perceive as more satisfying or useful than existing methods. In addition, by investigating news recommender systems in their entirety, rather than their individual components in solitude, we may form a better understanding of the interplay between components. Finally, taking different perspectives of news recommender systems in various contexts may reveal opportunities for interdisciplinary research, which may be the key to human-centered news recommender systems.

References

[1] Sean M McNee, John Riedl, and Joseph A Konstan. **Being accurate is not enough: how accuracy metrics have hurt recommender systems**. In *CHI'06 extended abstracts on Human factors in computing systems*, pages 1097–1101, 2006. ii, 2, 26

[2] Cai-Nicolas Ziegler, Sean M McNee, Joseph A Konstan, and Georg Lausen. **Improving recommendation lists through topic diversification**. In *Proceedings of the 14th international conference on World Wide Web*, pages 22–32, 2005. ii, 2, 28, 30, 51

[3] Rebecca K Ratner, Barbara E Kahn, and Daniel Kahneman. **Choosing less-preferred experiences for the sake of variety**. *Journal of consumer research*, **26**(1):1–15, 1999. ii, 2, 28, 51

[4] Yuan Cao Zhang, Diarmuid Ó Séaghdha, Daniele Quercia, and Tamas Jambor. **Auralist: introducing serendipity into music recommendation**. In *Proceedings of the fifth ACM international conference on Web search and data mining*, pages 13–22, 2012. ii, 2, 26, 27, 28, 29, 30, 32, 33, 51, 57, 60, 61

[5] Fangzhao Wu, Ying Qiao, Jiun-Hung Chen, Chuhan Wu, Tao Qi, Jianxun Lian, Danyang Liu, Xing Xie, Jianfeng Gao, Winnie Wu, et al. **Mind: A large-scale dataset for news recommendation**. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3597–3606, 2020. ii, 7, 8, 11, 43, 44, 45, 47, 51, 65

[6] R Meinl. **Recommender Systems: The Most Valuable Application of Machine Learning (Part 1)**, 2020. v, 18

[7] Yuan Zhang, Xue Dong, Weijie Ding, Biao Li, Peng Jiang, and Kun Gai. **Divide and Conquer: Towards Better Embedding-based Retrieval**

for Recommender Systems From a Multi-task Perspective. *arXiv preprint arXiv:2302.02657*, 2023. v, 19, 20, 21

[8] Pearl Pu, Li Chen, and Rong Hu. **A user-centric evaluation framework for recommender systems**. In *Proceedings of the fifth ACM conference on Recommender systems*, pages 157–164, 2011. v, 2, 25, 26, 27, 30, 37, 57, 68

[9] Ahmed El-Kishky, Thomas Markovich, Kenny Leung, Frank Portman, and Aria Haghighi. **kNN-Embed: Locally Smoothed Embedding Mixtures For Multi-interest Candidate Retrieval**. *arXiv preprint arXiv:2205.06205*, 2022. x, 21

[10] Francesco Ricci, Lior Rokach, and Bracha Shapira. **Recommender systems: Techniques, applications, and challenges**. *Recommender Systems Handbook*, pages 1–35, 2021. x, 28, 87

[11] Thiago Silveira, Min Zhang, Xiao Lin, Yiqun Liu, and Shaoping Ma. **How good your recommender system is? A survey on evaluations in recommendation**. *International Journal of Machine Learning and Cybernetics*, **10**:813–831, 2019. 1, 29, 30, 32, 33, 54, 87

[12] Kirsten Swearingen and Rashmi Sinha. **Beyond algorithms: An HCI perspective on recommender systems**. In *ACM SIGIR 2001 workshop on recommender systems*, **13**, pages 1–11, 2001. 2, 5, 25, 26, 27, 30, 57, 68, 87

[13] **Ergonomics of human-system interaction-Part 210: Human-centred design for interactive systems**. 2019. 2

[14] Warren S McCulloch and Walter Pitts. **A logical calculus of the ideas immanent in nervous activity**. *The bulletin of mathematical biophysics*, **5**:115–133, 1943. 3

[15] Oxford Languages. **Oxford Languages and Google-English. Languages**, 2021. 3

[16] Andrew D Selbst, Danah Boyd, Sorelle A Friedler, Suresh Venkatasubramanian, and Janet Vertesi. **Fairness and abstraction in sociotechnical systems**. In *Proceedings of the conference on fairness, accountability, and transparency*, pages 59–68, 2019. 3, 4, 5

[17] Chuhan Wu, Fangzhao Wu, Yongfeng Huang, and Xing Xie. **Personalized news recommendation: Methods and Challenges**. *ACM Transactions on Information Systems*, **41**(1):1–50, 2023. 3, 11, 12, 13, 14, 16, 63, 65

[18] Gediminas Adomavicius and Alexander Tuzhilin. **Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions**. *IEEE transactions on knowledge and data engineering*, **17**(6):734–749, 2005. 11

[19] Shaina Raza and Chen Ding. **News recommender system: a review of recent progress, challenges, and opportunities**. *Artificial Intelligence Review*, pages 1–52, 2022. 11, 12, 27, 44, 45

[20] Chuhan Wu, Fangzhao Wu, Mingxiao An, Jianqiang Huang, Yongfeng Huang, and Xing Xie. **Neural news recommendation with attentive multi-view learning**. *arXiv preprint arXiv:1907.05576*, 2019. 11, 46, 49, 51, 90, 91

[21] Chuhan Wu, Fangzhao Wu, Suyu Ge, Tao Qi, Yongfeng Huang, and Xing Xie. **Neural news recommendation with multi-head self-attention**. 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)*, pages 6389–6394, 2019. 11, 14, 46, 49, 51, 90

[22] Chuhan Wu, Fangzhao Wu, Mingxiao An, Yongfeng Huang, and Xing Xie. **Neural news recommendation with topic-aware news representation**. In *Proceedings of the 57th Annual meeting of the association for computational linguistics*, pages 1154–1159, 2019. 11, 49, 90

[23] Chuhan Wu, Fangzhao Wu, Mingxiao An, Jianqiang Huang, Yongfeng Huang, and Xing Xie. **NPA: neural news recommendation with personalized attention**. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 2576–2584, 2019. 11, 46

[24] Budi Juarto and Abba Suganda Girsang. **Neural collaborative with sentence BERT for news recommender system**. *JOIV: International Journal on Informatics Visualization*, **5**(4):448–455, 2021. 11

[25] Yongye Qian, Pengpeng Zhao, Zhixu Li, Junhua Fang, Lei Zhao, Victor S Sheng, and Zhiming Cui. **Interaction graph neural network for news**

recommendation. In *Web Information Systems Engineering–WISE 2019: 20th International Conference, Hong Kong, China, November 26–30, 2019, Proceedings 20*, pages 599–614. Springer, 2019. 11

[26] Linmei Hu, Chen Li, Chuan Shi, Cheng Yang, and Chao Shao. **Graph neural news recommendation with long-term and short-term interest modeling**. *Information Processing & Management*, **57**(2):102142, 2020. 11, 16

[27] TYSS Santosh, Avirup Saha, and Niloy Ganguly. **MVL: Multi-view learning for news recommendation**. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1873–1876, 2020. 11

[28] Qi Zhang, Jingjie Li, Qinglin Jia, Chuyuan Wang, Jieming Zhu, Zhaowei Wang, and Xiuqiang He. **UNBERT: User-News Matching BERT for News Recommendation.** In *IJCAI*, pages 3356–3362, 2021. 11

[29] Özlem Özgöbek, Jon Atle Gulla, and Riza Cenk Erdur. **A Survey on Challenges and Methods in News Recommendation.** In *WEBIST (2)*, pages 278–285, 2014. 12

[30] Mozhgan Karimi, Dietmar Jannach, and Michael Jugovac. **News recommender systems–Survey and roads ahead**. *Information Processing & Management*, **54**(6):1203–1227, 2018. 12, 44

[31] Miaomiao Li and Licheng Wang. **A survey on personalized news recommendation technology**. *IEEE Access*, **7**:145861–145879, 2019. 12

[32] Chong Feng, Muzammil Khan, Arif Ur Rahman, and Arshad Ahmad. **News recommendation systems-accomplishments, challenges & future directions**. *IEEE Access*, **8**:16702–16725, 2020. 12

[33] Marwa Hussien Mohamed, Mohamed Helmy Khafagy, and Mohamed Hasan Ibrahim. **Recommender systems challenges and solutions survey**. In *2019 international conference on innovative trends in computer engineering (ITCE)*, pages 149–155. IEEE, 2019. 12

[34] Ji Yang, Xinyang Yi, Derek Zhiyuan Cheng, Lichan Hong, Yang Li, Simon Xiaoming Wang, Taibai Xu, and Ed H Chi. **Mixed negative sampling for**

learning two-tower neural networks in recommendations. In *Companion Proceedings of the Web Conference 2020*, pages 441–447, 2020. 12, 16

[35] Bo Song, Xin Yang, Yi Cao, and Congfu Xu. **Neural collaborative ranking**. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pages 1353–1362, 2018. 12

[36] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. **Neural collaborative filtering**. In *Proceedings of the 26th international conference on world wide web*, pages 173–182, 2017. 12, 16

[37] Jinpeng Wang, Jieming Zhu, and Xiuqiang He. **Cross-batch negative sampling for training two-tower recommenders**. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1632–1636, 2021. 12

[38] Paul Covington, Jay Adams, and Emre Sargin. **Deep neural networks for youtube recommendations**. In *Proceedings of the 10th ACM conference on recommender systems*, pages 191–198, 2016. 12, 17, 19

[39] Xinyang Yi, Ji Yang, Lichan Hong, Derek Zhiyuan Cheng, Lukasz Heldt, Aditee Kumthekar, Zhe Zhao, Li Wei, and Ed Chi. **Sampling-bias-corrected neural modeling for large corpus item recommendations**. In *Proceedings of the 13th ACM Conference on Recommender Systems*, pages 269–277, 2019. 12, 15, 16, 17, 19, 20

[40] Chuhan Wu, Fangzhao Wu, Tao Qi, Qi Liu, Xuan Tian, Jie Li, Wei He, Yongfeng Huang, and Xing Xie. **Feedrec: News feed recommendation with various user feedbacks**. In *Proceedings of the ACM Web Conference 2022*, pages 2088–2097, 2022. 14

[41] Danyang Liu, Jianxun Lian, Shiyin Wang, Ying Qiao, Jiun-Hung Chen, Guangzhong Sun, and Xing Xie. **KRED: Knowledge-aware document representation for news recommendations**. In *Proceedings of the 14th ACM Conference on Recommender Systems*, pages 200–209, 2020. 14

[42] Chuhan Wu, Fangzhao Wu, Yongfeng Huang, and Xing Xie. **Neural news recommendation with negative feedback**. *CCF Transactions on Pervasive Computing and Interaction*, **2**:178–188, 2020. 16

[43] Lemei Zhang, Peng Liu, and Jon Atle Gulla. **A deep joint network for session-based news recommendations with contextual augmentation**. In *Proceedings of the 29th on Hypertext and Social Media*, pages 201–209. 2018. 16

[44] Chuhan Wu, Fangzhao Wu, Tao Qi, and Yongfeng Huang. **Sentirec: Sentiment diversity-aware neural news recommendation**. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pages 44–53, 2020. 16

[45] Jiaqi Ma, Zhe Zhao, Xinyang Yi, Ji Yang, Minmin Chen, Jiaxi Tang, Lichan Hong, and Ed H Chi. **Off-policy learning in two-stage recommender systems**. In *Proceedings of The Web Conference 2020*, pages 463–473, 2020. 17

[46] Zhe Zhao, Lichan Hong, Li Wei, Jilin Chen, Aniruddh Nath, Shawn Andrews, Aditee Kumthekar, Maheswaran Sathiamoorthy, Xinyang Yi, and Ed Chi. **Recommending what video to watch next: a multitask ranking system**. In *Proceedings of the 13th ACM Conference on Recommender Systems*, pages 43–51, 2019. 17, 88

[47] Fedor Borisyuk, Krishnaram Kenthapadi, David Stein, and Bo Zhao. **CaSMoS: A framework for learning candidate selection models over structured queries and documents**. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 441–450, 2016. 17

[48] Chantat Eksombatchai, Pranav Jindal, Jerry Zitao Liu, Yuchen Liu, Rahul Sharma, Charles Sugnet, Mark Ulrich, and Jure Leskovec. **Pixie: A system for recommending 3+ billion items to 200+ million users in real-time**. In *Proceedings of the 2018 world wide web conference*, pages 1775–1784, 2018. 17

[49] Christopher D Manning. *An introduction to information retrieval.* Cambridge university press, 2009. 19

[50] Zheng Liu, Yu Xing, Jianxun Lian, Defu Lian, Ziyao Li, and Xing Xie. **A novel user representation paradigm for making personalized candidate retrieval**. *arXiv preprint arXiv:1907.06323*, 2019. 19, 20

[51] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, et al. **Wide & deep learning for recommender systems**. In *Proceedings of the 1st workshop on deep learning for recommender systems*, pages 7–10, 2016. 19

[52] Ruoxi Wang, Bin Fu, Gang Fu, and Mingliang Wang. **Deep & cross network for ad click predictions**. In *Proceedings of the ADKDD'17*, pages 1–7. 2017. 19

[53] Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He. **DeepFM: a factorization-machine based neural network for CTR prediction**. *arXiv preprint arXiv:1703.04247*, 2017. 19

[54] Jianxun Lian, Xiaohuan Zhou, Fuzheng Zhang, Zhongxia Chen, Xing Xie, and Guangzhong Sun. **xdeepfm: Combining explicit and implicit feature interactions for recommender systems**. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 1754–1763, 2018. 19

[55] Jing Zhao, Jingya Wang, Madhav Sigdel, Bopeng Zhang, Phuong Hoang, Mengshu Liu, and Mohammed Korayem. **Embedding-based Recommender System for Job to Candidate Matching on Scale**. *arXiv preprint arXiv:2107.00221*, 2021. 19, 20

[56] Jeff Johnson, Matthijs Douze, and Hervé Jégou. **Billion-scale similarity search with GPUs**. *IEEE Transactions on Big Data*, **7**(3):535–547, 2019. 19, 50, 60, 61, 70

[57] Kohei Sugawara, Hayato Kobayashi, and Masajiro Iwasaki. **On approximately searching for similar word embeddings**. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2265–2275, 2016. 19

[58] Martin Aumüller, Erik Bernhardsson, and Alexander Faithfull. **ANN-Benchmarks: A benchmarking tool for approximate nearest neighbor algorithms**. *Information Systems*, **87**:101374, 2020. 19

[59] Tian Wang, Yuri M Brovman, and Sriganesh Madhvanath. **Personalized embedding-based e-commerce recommendations at ebay**. *arXiv preprint arXiv:2102.06156*, 2021. 20

[60] Jonathan L Herlocker, Joseph A Konstan, Loren G Terveen, and John T Riedl. **Evaluating collaborative filtering recommender systems**. *ACM Transactions on Information Systems (TOIS)*, **22**(1):5–53, 2004. 23, 24

[61] Himan Abdollahpouri, Edward C Malthouse, Joseph A Konstan, Bamshad Mobasher, and Jeremy Gilbert. **Toward the next generation of news recommender systems**. In *Companion proceedings of the web conference 2021*, pages 402–406, 2021. 24

[62] K Schrøder. **What do news readers really want to read about? How relevance works for news audiences**. *Digital News Project*, (Feb 2019), 2019. 24

[63] Stina Bengtsson. **The relevance of digital news: Themes, scales and temporalities**. *Digital Journalism*, pages 1–19, 2022. 24

[64] S Shyam Sundar. **Exploring receivers' criteria for perception of print and online news**. *Journalism & Mass Communication Quarterly*, **76**(2):373–386, 1999. 24, 25

[65] Gabriele Sottocornola, Panagiotis Symeonidis, and Markus Zanker. **Session-based news recommendations**. In *Companion Proceedings of the The Web Conference 2018*, pages 1395–1399, 2018. 27

[66] Michele Trevisiol, Luca Maria Aiello, Rossano Schifanella, and Alejandro Jaimes. **Cold-start news recommendation with domain-dependent browse graph**. In *Proceedings of the 8th ACM Conference on Recommender systems*, pages 81–88, 2014. 27

[67] Zeshan Fayyaz, Mahsa Ebrahimian, Dina Nawara, Ahmed Ibrahim, and Rasha Kashef. **Recommendation systems: Algorithms, challenges, metrics, and business opportunities**. *applied sciences*, **10**(21):7748, 2020. 27

[68] Shikha Agarwal and Archana Singhal. **Handling skewed results in news recommendations by focused analysis of semantic user profiles**. In *2014 International Conference on Reliability Optimization and Information Technology (ICROIT)*, pages 74–79. IEEE, 2014. 27

[69] Hao Ma, Xueqing Liu, and Zhihong Shen. **User fatigue in online news recommendation**. In *Proceedings of the 25th International Conference on World Wide Web*, pages 1363–1372, 2016. 27, 28

[70] Mathias Jesse, Christine Bauer, and Dietmar Jannach. **Intra-list similarity and human diversity perceptions of recommendations: the details matter**. *User Modeling and User-Adapted Interaction*, pages 1–34, 2022. 28, 29, 55, 56

[71] Swapneel Kalpesh Sheth, Jonathan Schaffer Bell, Nipun Arora, and Gail E Kaiser. **Towards diversity in recommendations using social networks**. 2011. 28, 54

[72] Matevž Kunaver and Tomaž Požrl. **Diversity in recommender systems–A survey**. *Knowledge-based systems*, **123**:154–162, 2017. 29

[73] Daniel M Fleder and Kartik Hosanagar. **Recommender systems and their impact on sales diversity**. In *Proceedings of the 8th ACM conference on Electronic commerce*, pages 192–199, 2007. 29

[74] Saúl Vargas, Linas Baltrunas, Alexandros Karatzoglou, and Pablo Castells. **Coverage, redundancy and size-awareness in genre diversity for recommender systems**. In *Proceedings of the 8th ACM Conference on Recommender systems*, pages 209–216, 2014. 29

[75] Charles LA Clarke, Maheedhar Kolla, Gordon V Cormack, Olga Vechtomova, Azin Ashkan, Stefan Büttcher, and Ian MacKinnon. **Novelty and diversity in information retrieval evaluation**. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 659–666, 2008. 29

[76] Saúl Vargas. **Novelty and diversity enhancement and evaluation in recommender systems and information retrieval**. In *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*, pages 1281–1281, 2014. 29

[77] Makoto Nakatsuji, Yasuhiro Fujiwara, Akimichi Tanaka, Toshio Uchiyama, Ko Fujimura, and Toru Ishida. **Classical music for rock fans? Novel recommendations for expanding user interests**. In *Proceedings of*

the 19th ACM international conference on Information and knowledge management, pages 949–958, 2010. 30

[78] Tao Zhou, Zoltán Kuscsik, Jian-Guo Liu, Matúš Medo, Joseph Rushton Wakeling, and Yi-Cheng Zhang. **Solving the apparent diversity-accuracy dilemma of recommender systems**. *Proceedings of the National Academy of Sciences*, **107**(10):4511–4515, 2010. 30, 44, 58, 59

[79] Pablo Castells, Saúl Vargas, and Jun Wang. **Novelty and diversity metrics for recommender systems: choice, discovery and relevance**. 2011. 31, 44, 58, 59

[80] Abhijnan Chakraborty, Saptarshi Ghosh, Niloy Ganguly, and Krishna P Gummadi. **Optimizing the recency-relevance-diversity trade-offs in non-personalized news recommendations**. *Information Retrieval Journal*, **22**:447–475, 2019. 31

[81] Andrii Maksai, Florent Garcin, and Boi Faltings. **Predicting online performance of news recommender systems through richer evaluation metrics**. In *Proceedings of the 9th ACM Conference on Recommender Systems*, pages 179–186, 2015. 32

[82] Reza Jafari Ziarani and Reza Ravanmehr. **Serendipity in recommender systems: a systematic literature review**. *Journal of Computer Science and Technology*, **36**:375–396, 2021. 32

[83] Marco De Gemmis, Pasquale Lops, Giovanni Semeraro, and Cataldo Musto. **An investigation on the serendipity problem in recommender systems**. *Information Processing & Management*, **51**(5):695–717, 2015. 32

[84] Ivy Jain and Hitesh Hasija. **An effective approach for providing diverse and serendipitous recommendations**. In *Information Systems Design and Intelligent Applications: Proceedings of Third International Conference INDIA 2016, Volume 3*, pages 11–18. Springer, 2016. 33

[85] Tomoko Murakami, Koichiro Mori, and Ryohei Orihara. **Metrics for evaluating the serendipity of recommendation lists**. In *New Frontiers in Artificial Intelligence: JSAI 2007 Conference and Workshops, Miyazaki, Japan, June 18-22, 2007, Revised Selected Papers*, pages 40–46. Springer, 2008. 33

[86] Mouzhi Ge, Carla Delgado-Battenfeld, and Dietmar Jannach. **Beyond accuracy: evaluating recommender systems by coverage and serendipity**. In *Proceedings of the fourth ACM conference on Recommender systems*, pages 257–260, 2010. 33

[87] Panagiotis Adamopoulos and Alexander Tuzhilin. **On unexpectedness in recommender systems: Or how to better expect the unexpected**. *ACM Transactions on Intelligent Systems and Technology (TIST)*, **5**(4):1–32, 2014. 34

[88] Marius Kaminskas and Derek Bridge. **Measuring surprise in recommender systems**. In *Proceedings of the workshop on recommender systems evaluation: dimensions and design (Workshop programme of the 8th ACM conference on recommender systems)*. Citeseer, 2014. 34

[89] Guido van Capelleveen, Chintan Amrit, Devrim Murat Yazan, and Henk Zijm. **The recommender canvas: A model for developing and documenting recommender system design**. *Expert systems with applications*, **129**:97–117, 2019. 37

[90] Engin Bozdag and Jeroen Van Den Hoven. **Breaking the filter bubble: democracy and design**. *Ethics and information technology*, **17**:249–265, 2015. 37, 42

[91] Michael A Beam. **Automating the news: How personalized news recommender system design choices impact news reception**. *Communication Research*, **41**(8):1019–1041, 2014. 37

[92] Heng Xu, Xin Robert Luo, John M Carroll, and Mary Beth Rosson. **The personalization privacy paradox: An exploratory study of decision making process for location-aware marketing**. *Decision support systems*, **51**(1):42–52, 2011. 37

[93] Sean M McNee, Shyong K Lam, Joseph A Konstan, and John Riedl. **Interfaces for eliciting new user preferences in recommender systems**. In *User Modeling 2003: 9th International Conference, UM 2003 Johnstown, PA, USA, June 22–26, 2003 Proceedings 9*, pages 178–187. Springer, 2003. 37

[94] Jon Espen Ingvaldsen, Jon Atle Gulla, and Özlem Özgöbek. **User Controlled News Recommendations.** In *IntRS@ RecSys*, pages 45–48, 2015. 37

[95] Dietmar Jannach. **Multi-Objective Recommender Systems: Survey and Challenges**. *arXiv preprint arXiv:2210.10309*, 2022. 38, 39, 40

[96] Gerhard Fischer. **Context-aware systems: the 'right' information, at the 'right' time, in the 'right' place, in the 'right' way, to the 'right' person**. In *Proceedings of the international working conference on advanced visual interfaces*, pages 287–294, 2012. 38

[97] Benjamin Uwe Kille. *On context-aware news recommender systems*. PhD thesis, Technische Universität Berlin, 2021. 38

[98] Iván Cantador, Pablo Castells, et al. **Semantic contextualisation in a news recommender system**. In *Workshop on Context-Aware Recommender Systems (CARS 2009)*, **1068**, pages 19–25. Citeseer, 2009. 38

[99] Toon De Pessemier, Cédric Courtois, Kris Vanhecke, Kristin Van Damme, Luc Martens, and Lieven De Marez. **A user-centric evaluation of context-aware recommendations for a mobile news service**. *Multimedia Tools and Applications*, **75**:3323–3351, 2016. 38

[100] Dirk Bollen, Bart P Knijnenburg, Martijn C Willemsen, and Mark Graus. **Understanding choice overload in recommender systems**. In *Proceedings of the fourth ACM conference on Recommender systems*, pages 63–70, 2010. 38

[101] Yuyu Zhang, Hanjun Dai, Chang Xu, Jun Feng, Taifeng Wang, Jiang Bian, Bin Wang, and Tie-Yan Liu. **Sequential click prediction for sponsored search with recurrent neural networks**. In *Proceedings of the AAAI Conference on Artificial Intelligence*, **28**, 2014. 38

[102] Huifeng Guo, Jinkai Yu, Qing Liu, Ruiming Tang, and Yuzhou Zhang. **PAL: a position-bias aware learning framework for CTR prediction in live recommender systems**. In *Proceedings of the 13th ACM Conference on Recommender Systems*, pages 452–456, 2019. 38

[103] Yashar Deldjoo, Markus Schedl, Balázs Hidasi, Yinwei Wei, and Xiangnan He. **Multimedia recommender systems: Algorithms and challenges**. In *Recommender systems handbook*, pages 973–1014. Springer, 2021. 39

[104] Emily Bender. **The# benderrule: On naming the languages we study and why it matters**. *The Gradient*, **14**, 2019. 39

[105] Dietmar Jannach and Gediminas Adomavicius. **Recommendations with a purpose**. In *Proceedings of the 10th ACM conference on recommender systems*, pages 7–10, 2016. 39, 40

[106] Natali Helberger. **On the democratic role of news recommenders**. *Digital Journalism*, **7**(8):993–1012, 2019. 40, 41, 42

[107] Silvia Milano, Mariarosaria Taddeo, and Luciano Floridi. **Recommender systems and their ethical challenges**. *Ai & Society*, **35**:957–967, 2020. 41, 42

[108] Sorelle A Friedler, Carlos Scheidegger, and Suresh Venkatasubramanian. **On the (im) possibility of fairness**. *arXiv preprint arXiv:1609.07236*, 2016. 41

[109] Guusje Juijn. *Perceived Algorithmic Fairness using Organizational Justice Theory: an Empirical Case Study on Algorithmic Hiring*. Master's thesis, 2023. 42

[110] Ansgar Koene, Elvira Perez, Christopher James Carter, Ramona Statache, Svenja Adolphs, Claire O'Malley, Tom Rodden, and Derek McAuley. **Ethics of personalized information filtering**. In *Internet Science: Second International Conference, INSCI 2015, Brussels, Belgium, May 27-29, 2015, Proceedings 2*, pages 123–132. Springer, 2015. 42

[111] Jaron Harambam, Natali Helberger, and Joris Van Hoboken. **Democratizing algorithmic news recommenders: how to materialize voice in a technologically saturated media ecosystem**. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, **376**(2133):20180088, 2018. 42

[112] Natali Helberger, Kari Karppinen, and Lucia D'acunto. **Exposure diversity as a design principle for recommender systems**. *Information, Communication & Society*, **21**(2):191–207, 2018. 42

[113] Urbano Reviglio. **Serendipity by design? How to turn from diversity exposure to diversity experience to face filter bubbles in social media**. In

Internet Science: 4th International Conference, INSCI 2017, Thessaloniki, Greece, November 22-24, 2017, Proceedings 4, pages 281–300. Springer, 2017. 42

[114] Nan Wang, Hongning Wang, Yiling Jia, and Yue Yin. **Explainable recommendation via multi-task learning in opinionated text data**. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 165–174, 2018. 42

[115] Mats Mulder, Oana Inel, Jasper Oosterman, and Nava Tintarev. **Operationalizing framing to support multiperspective recommendations of opinion pieces**. In *Proceedings of the 2021 ACM conference on fairness, accountability, and transparency*, pages 478–488, 2021. 42

[116] Tao Qi, Fangzhao Wu, Chuhan Wu, Yongfeng Huang, and Xing Xie. **Privacy-preserving news recommendation model learning**. *arXiv preprint arXiv:2003.09592*, 2020. 42

[117] Lucien Heitz, Juliane A Lischka, Alena Birrer, Bibek Paudel, Suzanne Tolmeijer, Laura Laugwitz, and Abraham Bernstein. **Benefits of diverse news recommendations for democracy: A user study**. *Digital Journalism*, **10**(10):1710–1730, 2022. 42

[118] Benjamin Kille, Frank Hopfgartner, Torben Brodt, and Tobias Heintz. **The plista dataset**. In *Proceedings of the 2013 international news recommender systems workshop and challenge*, pages 16–23, 2013. 44

[119] Jon Atle Gulla, Lemei Zhang, Peng Liu, Özlem Özgöbek, and Xiaomeng Su. **The adressa dataset for news recommendation**. In *Proceedings of the international conference on web intelligence*, pages 1042–1048, 2017. 44

[120] Wei Chu, Seung-Taek Park, Todd Beaupre, Nitin Motgi, Amit Phadke, Seinjuti Chakraborty, and Joe Zachariah. **A case study of behavior-driven conjoint analysis on Yahoo! Front Page Today module**. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1097–1104, 2009. 44

[121] Mingxiao An, Fangzhao Wu, Chuhan Wu, Kun Zhang, Zheng Liu, and Xing Xie. **Neural news recommendation with long-and short-term user**

representations. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 336–345, 2019. 46, 49, 51, 90

[122] Niran A Abdulhussein and Ahmed J Obaid. **User recommendation system based on MIND dataset**. *arXiv preprint arXiv:2209.06131*, 2022. 47

[123] Qin Ruan. **Mind News Recommendation Technical Report**. 2020. 47

[124] Microsoft. **Recommenders**. `https://github.com/microsoft/recommenders`, 2023. 49, 50

[125] Yuting Ning. **MIND**. `https://github.com/nnnyt/MIND`, 2020. 50, 90

[126] Shaina Raza and Chen Ding. **Deep dynamic neural network to trade-off between accuracy and diversity in a news recommender system**. *arXiv preprint arXiv:2103.08458*, 2021. 54

[127] Guanjie Zheng, Fuzheng Zhang, Zihan Zheng, Yang Xiang, Nicholas Jing Yuan, Xing Xie, and Zhenhui Li. **DRN: A deep reinforcement learning framework for news recommendation**. In *Proceedings of the 2018 world wide web conference*, pages 167–176, 2018. 54

[128] Sanne Vrijenhoek, Mesut Kaya, Nadia Metoui, Judith Möller, Daan Odijk, and Natali Helberger. **Recommenders with a mission: assessing diversity in news recommendations**. In *Proceedings of the 2021 conference on human information interaction and retrieval*, pages 173–183, 2021. 64

[129] Jeffrey Pennington, Richard Socher, and Christopher D Manning. **Glove: Global vectors for word representation**. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014. 90

8

Appendix

8.1 Accuracy Metrics

In offline evaluation, accuracy-metrics are used to asses the utility (sometimes referred to as relevance, usefulness, recommendation value or satisfaction [11]) of recommendations for users. Ricci et al. [10] argue that utility may be defined as the value that users receives from a certain recommendation. Hence, accuracy metrics are used as a (inherently limited) proxy for which items are more valuable to users than others. Although accuracy is no longer the only objective of researchers and practitioners, it may still be one of the primary factors contributing to user satisfaction [12]. Therefore, the accuracy of recommender systems should not suffer in the pursuit of beyond-accuracy objectives. There are many metrics available for evaluating the accuracy of recommendations, but common metrics are the AUC-score, Mean Reciprocal Rank (MMR), and Normalized Discounted Cumulative Gain (nDCG), precision and recall, which we will briefly elaborate on below.

8.1.1 AUC-score

The AUC-score represents the possibility that a random positive sample is ranked higher than a random negative sample. The AUC score ranges from 0 to 1. If the recommender system ranks all positive samples higher than any negative sample, the AUC-score will be 1. The AUC-score works well for evaluating recommender systems based on implicit feedback, because it only takes into account the relative position of recommendations, rather than the absolute values of the predictions. Let $\{x_1, ..., x_n\} \in R_u$ be the set of positive samples in the list of recommended items R_u and let $\{y_1, ..., y_m\}$ be the set of negative samples in the list of recommended items. Than the AUC-score can be estimated using the Mann-Whitney U statistic as follows:

$$U = \sum_{i=1}^{n} \sum_{j=1}^{m} S(x_i, y_j), \tag{8.1}$$

with

$$S(x,y) = \begin{cases} 1, & \text{if } x \text{ is ranked higher than } y \\ \frac{1}{2}, & \text{if } x \text{ is ranked equal to } y \\ 0, & \text{if } x \text{ is ranked lower than } y \end{cases} \tag{8.2}$$

Using the Mann-Whitney U statistic, the AUC-score can be calculated as:

$$AUC = \frac{U}{n * m} \tag{8.3}$$

8.1.2 Mean Reciprocal Rank

The Mean Reciprocal Rank (MRR) is used to evaluate the average rank of the highest-ranked positive sample in the list of recommendations. For a single list of recommendations, the reciprocal rank is $\frac{1}{rank}$ where $rank$ is the position of the highest-ranked positive sample. For multiple recommendation queries Q, the MRR is given by:

$$MRR = \frac{1}{Q} \sum_{i=1}^{Q} \frac{1}{rank_i} \tag{8.4}$$

The position of items in a list of recommendations has been shown to have a large impact on the likelihood of users clicking on the item [46]. The higher an item is ranked, the more likely users are to click on the item. Therefore, the MRR is an important metric in evaluating the accuracy of recommender systems.

8.1.3 Normalized Discounted Cumulative Gain

The Normalized Discounted Cumulative Gain (NDCG) metric is widely used in recommender systems to evaluate the effectiveness of ranking algorithms. It takes into account both the relevance of recommended items and their position in the ranked list. NDCG assigns higher scores to systems that not only rank relevant items higher but also promote them towards the top of the list. By incorporating the concept of discounting, NDCG reflects the diminishing value of relevance as the position of an item increases. The NDCG is a normalized version of the Discounted Cumulative Gain (DCG) metric to facilitate comparisons across different datasets and scenarios. The DCG and NDCG can be calculated as follow:

$$DCG_p = \sum_{i=1}^{p} \frac{rel_i}{\log_2(i+1)}, \tag{8.5}$$

where rel_i is the graded relevance of the item at position i ($rel_i = 1$ if the user clicked on item i and $rel_i = 0$ if the user did not click on item i in the case of implicit click feedback). The NDCG is computed as:

$$NDCG_p = \frac{DCG_p}{IDCG_i}, \tag{8.6}$$

where IDCG is the Ideal Discounted Cumulative Gain:

$$IDCG_p = \sum_{i=1}^{|REL_p|} \frac{rel_i}{\log_2(i+1)}, \tag{8.7}$$

and REL_p represent the ideal list of recommendations ordered by their relevance. The DCG and NDCG are generally evaluated over the top p positions in a list of recommendations, e.g. $p = 5$ or $p = 10$.

8.1.4 Precision and Recall

Other important metrics used for evaluating recommendations are precision, recall and the F1-score. Precision and recall can either be used to evaluate the candidate retrieval stage or the ranking stage. In the candidate retrieval stage, precision is calculated as the proportion of relevant candidates retrieved out of the whole set of candidates. Recall in the candidate retrieval stage is the proportion of relevant candidates retrieved from all relevant candidates in the catalog. That is:

$$precision = \frac{TP}{TP+FP}, \tag{8.8}$$

$$recall = \frac{TP}{TP+FN}, \tag{8.9}$$

with TP the True Positives, FP the False Positives, and FN the False Negatives. The precision and recall metric can both be adjusted slightly for evaluating the ranking stage. In that case, precision and recall are evaluated at the k recommendations that are ranked highest in the list:

$$precision@k = \frac{\text{\# of relevant items @k}}{k}, \tag{8.10}$$

$$recall@k = \frac{\#\text{ of relevant items @k}}{\#\text{ of relevant items in full list of recommendations}}. \tag{8.11}$$

In both cases, the F1-score is defined as the harmonic mean of the precision and recall:

$$F_1 = 2\frac{precision \cdot recall}{precision + recall}. \tag{8.12}$$

For recommender systems that leverage explicit feedback in the form of ratings, common metrics are: Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and Mean Squared Error (MSE).

8.2 Experimental Settings Ranking Models

In this section I will describe the details of the implementation and hyperparameters of the NRMS [21], LSTUR [121], NAML [20] and TANR [22] model introduced in Section 5.2. The experimental settings follow the original works, unless specified otherwise.

For the NRMS, NAML, and TANR models, the word embeddings are 300-dimensional and initialized by the pre-trained GloVe embedding [129], as done in previous works [20; 21; 22]. The LSTUR model also uses GloVe to create pre-trained word embeddings, but the embeddings are 200-dimensional, similar to previous work [121]. Both the NRMS and TANR model only use the title of each news-article as input feature. The LSTUR model also uses the category and subcategory as input features. The NAML model has the most input features, as it uses the title, category, subcategory, and abstract of news-articles. In all cases the length of the title is trimmed to be 30 tokens long. For the NAML model the length of the abstract is trimmed to be 100 tokens at maximum.

NRMS In this thesis, the same implementation details as the original work of Wu et al. [21] is followed. Therefore, the self-attention network in the news-encoder and the user-encoder has 16 heads in both cases and each head has a 16-dimensional output. The additive query vectors are 200-dimensional. The negative sampling ratio K is 4 and the batch size is 64. Dropout is applied to each layer in the model with a rate of 0.2 to mitigate overfitting. Adam is used to optimize the model and the learning rate is set to 0.01.

The first layer in the model, the embedding layer, matches each unique word in the title to its corresponding GloVe embedding. In the original implementation of Ning [125] every unique word in both the training and test set is matched to a GloVe embedding. This requires extracting all unique words from news articles present in both the training and test set up front, to create a vocabulary of all words that might be present during training or testing. One might argue that this can be considered test set leakage, because normally if an unknown word occurs in the test set, it wouldn't be possible to match it to its respective GloVe-embedding. Therefore, in the implementation of this work, the vocabulary of words is only based on words present in the news articles of the training set. In addition, instead of creating a full vocabulary of all unique words, this implementation uses a vocabulary with a fixed size of the top 20.000 most frequent words in the training set. This has the added benefit that the vocabulary has the same size regardless of whether the model is used in training, validation or testing, and did not seem to hinder performance.

LSTUR The number of filters in the CNN is 400, the window size is 3, the stride is set to 1 and zero-padding is applied. The attention layer after the CNN is 200-dimensional. A dropout of 20% is applied to each layer. Both the category and subcategory are encoded to 100-dimensional vectors. The long-term user representation is masked with a probability $p = 0.5$, which yields the best performance according to the experiments conducted by the authors. Adam is used to optimize the model and the learning rate is set to 0.01. In the original paper of the LSTUR model, the authors used a batch-size of 400, but this put to much strain on computational resource, hence the batch-size here is set to 64.

NAML The embedding dimensions of the category embedding are set to 100. The number of filters in the CNN is 400, the window size is 3, the stride is set to 1 and zero-padding is applied. The attention layer after the CNN is 200-dimensional. A dropout of 20% is applied to each layer. Both the category and subcategory are encoded by dense layers with a dimension of 400. Adam is used to optimize the model and the learning rate is set to 0.01. The negative sampling rate was set to 4. The only alteration in the hyperparameters compared to the work of Wu et al. [20] is that the batch size is reduced to 64, instead of 100, in order not to overflow the available computational memory.

TANR The number of filters in the CNN is 400, the window size is 3, the stride is set to 1 and zero-padding is applied. The attention layer after the CNN is 200-dimensional, and has a dropout rate of 0.2. Adam is used to optimize the model and the learning rate is set to 0.01.

8.3 Intra-list similarity using latent representations

<table>
<tr><th colspan="5">Example 1</th></tr>
<tr><th>News ID</th><th>Category</th><th>SubCategory</th><th>Title</th><th>Abstract</th></tr>
<tr><td>N69972</td><td>Lifestyle</td><td>lifestylepetsanimals</td><td>27 "Facts" About Animals You Have All Wrong</td><td>Spoiler alert: you may never order grilled octopus again. The post 27 "Facts" About Animals You Have All Wrong appeared first on Reader's Digest.</td></tr>
<tr><td>N126158</td><td>lifestyle</td><td>lifestyledidyouknow</td><td>50 Facts So Far-Fetched You Can't Help But Question Them</td><td>There's no doubt that the world is a weird place. Here are 75 weird facts about historical events, celebrities, and animals.</td></tr>
<tr><td>N51290</td><td>finance</td><td>finance-companies</td><td>32 facts about Ikea you probably didn't know</td><td>It's not just Ikea's furniture that has had a global impact</td></tr>
<tr><td>N60307</td><td>lifestyle</td><td>lifestyledidyouknow</td><td>30 Facts About Halloween No One Ever Told You</td><td>Find out the details about cool tidbits of info like how much Americans spend on Halloween and more with these 30 Halloween facts that no one ever told you.</td></tr>
<tr><td>N22668</td><td>finance</td><td>financenews</td><td>9 facts about felons' rights</td><td>Florida voters overwhelmingly approved the automatic restoration of rights for felons, but the process has been anything but automatic.</td></tr>
<tr><th colspan="5">Example 2</th></tr>
<tr><td>N115723</td><td>finance</td><td>finance-top-stocks</td><td>Wendy's Shares Fizzle as Analyst Pans Breakfast Plans</td><td>Wendy's plans to spend $20 million in a bid to gain a foothold in the lucrative breakfast market, a move some industry analysts are panning.</td></tr>
<tr><td>N94143</td><td>foodanddrink</td><td>foodnews</td><td>The Pioneer Woman's Frozen Food Line Will Save You From Any Holiday Stress</td><td>The mac and cheese is CRAZY good.</td></tr>
<tr><td>N92430</td><td>foodanddrink</td><td>newstrends</td><td>Local home-brew shop seeks actual craft-brewery license to up its game</td><td>A local home-brewing shop has applied to the Ohio Division of Liquor Control for a craft-brewery license that the shop's founder says will help boost his company's profile and its business.</td></tr>
<tr><td>N99533</td><td>video</td><td>lifestyle</td><td>Pie-eating contest is a secret cover for sweet homecoming</td><td>A pie-eating contest turned into messy hugs and kisses for these three boys and their Airman dad.</td></tr>
<tr><td>N4748</td><td>foodanddrink</td><td>newstrends</td><td>How to Carve a Turkey, According to Knife Experts</td><td>You've roasted the bird, it's absolutely perfect, and now all your guests are waiting for you to carve it. How to carve a turkey the right way depends entirely on whom you ask. But one thing's for sure: There's a subtle art and science to it, and...</td></tr>
</table>

Table 8.1: Examples of news recommendations with different ILS-scores