

Lars Kristian Holmedal Gjelstad, Eivind Fålun

**NTNU**  
Norwegian University of  
Science and Technology  
Faculty of Information Technology and Electrical  
Engineering  
Department of Computer Science

Lars Kristian Holmedal Gjelstad  
Eivind Fålun

# Explaining Neural Based News Recommender Systems

July 2021





Norwegian University of  
Science and Technology

# Explaining Neural Based News Recommender Systems

**Lars Kristian Holmedal Gjelstad**  
**Eivind Fålnun**

Computer Science

Submission date: July 2021

Supervisor: Jon-Atle Gulla

Co-supervisor: Peng Liu

Norwegian University of Science and Technology  
Department of Computer Science





## Abstract

The recent years have witnessed increased efforts in developing measures to aid users in navigating online services through recommender systems. These efforts are not unwarranted, as the utilization of such systems have demonstrated increased user engagement and satisfaction through relieving users from information overload.

Due to the increasing demand and commercial value of recommender systems, recent research on increasing their efficiency and accuracy have resulted in state-of-the art recommender techniques that implement methods from deep learning. Although these techniques increase the accuracy of the recommendations, their inherent complexity with high number of parameters have resulted in the recommendation engines being deemed as black boxes — as they providing little to no transparency to the recommendation process.

To address this, we propose ENSUS — a SHAP based model for explaining a deep learning based news recommender system through highlighting feature importance of input values. The model is based on a game theoretic approach known as Shapley values, where input values in a neural network are paired up in a game theoretic environment. The resulting feature importance reflects the contribution of each feature on the output — or in this case, the recommendation.

In addition, we propose a second approach to explanation that fully omit the black-box, and justifies the recommendations based on contextual similarities between knowledge objects, namely that of recently viewed news articles.

Our proposed methods are quantitatively evaluated through a user survey, through which we demonstrate that a neural based news recommender explained through highlighting feature importance drastically increases users perceived transparency. However, this does not imply an increase in trust, as our approach to justification performs equally as well in gaining the trust of users. This is an interesting discovery, as it demonstrates that omitting the black-box can indeed increase users trust in the recommendation process without the need for complex explanatory measures. Furthermore, this thesis provides evidence that our proposed method enables a user to influence future recommendations. Experiments are performed with two large datasets in both English and Norwegian to demonstrate the effectiveness of Shapley values in a commercial recommender system.



## Sammendrag

De siste årene har bevitnet økt innsats forbundet med utviklingen av verktøy for å assistere brukere i å navigere nettbaserte underholdningstjenester gjennom anbefalingssystemer. Denne innsatsen er ikke ubegrunnet, da det er demonstrert at slike systemer øker brukerengasjement og tilfredshet gjennom å begrense eksponeringen av informasjonsoverbelastning på nettbaserte underholdningstjenester.

Grunnet økt etterspørsel og kommersiell verdi, så har forskning på å effektivisere anbefalingssystemer ført til en ny generasjon toppmoderne anbefalingssystemer som benytter seg av dyp læring som den underliggende beslutningstakeren. Til tross for at disse metodene har økt effektiviteten til systemene, så øker dem samtidig kompleksiteten til anbefalingssystemet i bunn ved å innføre et enormt antall parametere. Dette har ført til at moderne anbefalingssystemer blir kalt *sorte bokser*, da de gir tilnærmet ingen innsikt eller forståelse for den underliggende anbefalingsprosessen.

For å imøtekomme disse problemene, foreslår vi ENSUS - en SHAP basert modell for å forklare nyhets anbefalingssystemer basert på dyp læring. Modellen er basert på en metode fra spillteori der attributter i et nevralt nettverk er sammenlignet i en simulert konkurranse, der forklaringer genereres ved å fortelle brukeren hvor mye hver attributt i datasettet bidrar til de endelige prediksjonene gjennom å sammenligne konkurransebidraget fra hver attributt.

I tillegg foreslår vi en metode som forsøker å rettferdiggjøre forklaringene ved å gå rundt den sorte boksen, og utelukkende se på kontekstuell likhet mellom historikken til leseren og den anbefalte artikkelen.

Metodene blir kvantitativt evaluert ved bruk av en brukerundersøkelse. Resultatene fra brukerundersøkelsen viser at ENSUS øker brukeres oppfattelse av gjennomsiktighet. Derimot viser undersøkelse at den ikke øker troverdighet fordi metoden for likhet presterer like bra på troverdighet. Videre viser eksperimentene i denne oppgaven at ENSUS tilrettelegger for at brukeren kan påvirke sine fremtidige anbefalinger ved å forklare systemet hvilke type nyhetskategorier som er ønsket.





## Preface

The following thesis constitutes the work of a master thesis in Computer Science at the Norwegian University of Science and Technology (NTNU), with the Department of Computer Science and Informatics (IDI).

We would like to thank our supervisor Prof. Jon Atle Gulla for his guidance and valuable feedback while supervising us during the research and development of this thesis. We would also like to thank our co-supervisor PhD Peng Liu for assisting us. Next, we would like to thank PhD Robindra Prabhu for introducing us to Shapley values and PhD Nils Barlaug for explaining and assisting us with the Shapley values library SHAP.

Lastly we extend our gratitude to our family, friends and colleagues for supporting us and assisting with proof-reading this thesis.



# Contents

<b>List of Figures</b>	<b>ix</b>
<b>List of Tables</b>	<b>xi</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Motivation . . . . .	1
1.2 Goals and Research Questions . . . . .	2
1.2.1 Research Questions (RQ)s . . . . .	3
1.3 Research Method . . . . .	4
1.4 Research Process . . . . .	4
1.4.1 Initial Literature Search . . . . .	4
1.4.2 Structured Literature Review Protocol . . . . .	5
1.4.3 Structured Literature Review . . . . .	6
1.5 Results . . . . .	7
1.6 Thesis Overview . . . . .	8
<b>2 Background Theory</b>	<b>9</b>
2.1 Recommending News Articles . . . . .	9
2.1.1 Characteristics of News Articles . . . . .	10
2.1.2 Challenges for Recommending News Articles . . . . .	11
2.2 Recommendation Paradigms . . . . .	12
2.2.1 Content-Based Filtering . . . . .	13
2.2.2 Collaborative Filtering . . . . .	14
2.2.3 Hybrid Systems . . . . .	16
2.2.4 Knowledge-Based Filtering . . . . .	16

2.3	Document Representations . . . . .	16
2.3.1	Traditional Word Representations . . . . .	16
2.3.2	Word Embeddings . . . . .	17
2.3.3	Pre-Trained Contextual Embeddings . . . . .	18
2.3.4	Sentence Embeddings . . . . .	19
2.4	Artificial Neural Networks . . . . .	20
2.4.1	Overview of Training a Neural Network . . . . .	21
2.4.2	Forward Propagation . . . . .	21
2.4.3	Learning with Gradient Descent . . . . .	22
2.4.4	Activation functions . . . . .	22
2.4.5	Output Functions . . . . .	23
2.4.6	Topologies . . . . .	23
2.4.7	Regularization . . . . .	25
2.4.8	Attention mechanisms . . . . .	26
2.5	Deep Learning in Recommender Systems . . . . .	27
2.5.1	Methods for Neural Recommender Systems . . . . .	28
2.5.2	Deep Learning in News Recommendation Systems . . . . .	29
2.6	Explainable Artificial Intelligence . . . . .	33
2.6.1	Methods for Explanation . . . . .	33
2.6.2	Post-Hoc Explainability . . . . .	34
2.6.3	Interpretable Models for Explainability . . . . .	35
<b>3</b>	<b>Taxonomy of Explanations in Recommender Systems</b>	<b>38</b>
3.1	Establishing Terminology . . . . .	39
3.2	Taxonomy Overview . . . . .	40
3.3	Information Sources . . . . .	41
3.3.1	User Preference and User Input . . . . .	42
3.3.2	Decision Inference Process . . . . .	43
3.3.3	Background and Complementary Information . . . . .	44
3.3.4	Alternatives and Their Features . . . . .	45
3.4	Presentation Styles . . . . .	46

3.4.1	Recommender Inspired Styles . . . . .	46
3.4.2	Feature Explanations . . . . .	48
3.4.3	Sentence Explanations . . . . .	48
3.4.4	Visual Explanations . . . . .	49
3.4.5	Hybrid Explanations . . . . .	49
3.5	Methods for Explaining Recommendations . . . . .	50
3.5.1	Matrix Factorization Models . . . . .	50
3.5.2	Topic Modelling . . . . .	51
3.5.3	Graph-based models for explainable recommendations . . . . .	52
3.5.4	Deep Learning for explainable recommendations . . . . .	52
3.5.5	Model Agnostic Methods . . . . .	53
3.6	Evaluating Explanations . . . . .	54
3.6.1	Means of Evaluation . . . . .	54
3.6.2	Metrics for Evaluation . . . . .	55
3.6.3	Levels of Explanations . . . . .	57
3.7	Summary . . . . .	58
<b>4</b>	<b>Related Work</b>	<b>60</b>
4.1	Explainable Recommendations . . . . .	60
4.1.1	Explainable News Recommendation . . . . .	60
4.2	Source and Presentation of Explanations . . . . .	61
4.2.1	Highlighting Feature Relevance . . . . .	61
4.2.2	Highlighting Similarity . . . . .	62
4.2.3	Highlighting Nearest Neighbours . . . . .	63
4.2.4	Highlighting Influence . . . . .	63
4.2.5	Combining Presentation Styles . . . . .	63
4.3	Methods for Explaining Recommender Systems . . . . .	64
4.3.1	Determining Feature Relevance . . . . .	64
4.3.2	Determining Similarity . . . . .	65
4.3.3	Determining Nearest Neighbours . . . . .	65
4.4	Evaluating Explanations . . . . .	66

4.4.1	User Studies . . . . .	66
4.4.2	Online Evaluation . . . . .	68
4.4.3	Offline Evaluation . . . . .	68
<b>5</b>	<b>Data</b>	<b>70</b>
5.1	Available Datasets . . . . .	70
5.2	The Adressa Dataset . . . . .	71
5.2.1	Characteristics . . . . .	71
5.2.2	Articles . . . . .	72
5.3	The MIND Dataset . . . . .	74
5.3.1	Characteristics . . . . .	74
5.3.2	Preprocessing . . . . .	75
5.4	Attribute Selection . . . . .	75
5.4.1	Items . . . . .	76
5.4.2	Users and Interactions . . . . .	76
5.4.3	Notable Observations . . . . .	77
<b>6</b>	<b>Method</b>	<b>78</b>
6.1	Conceptualizing the ENSUS Model . . . . .	78
6.2	Proposed ENSUS Model . . . . .	80
6.2.1	User Profile Generator . . . . .	80
6.2.2	Recommender Component . . . . .	81
6.2.3	Explanations Generator . . . . .	82
6.2.4	Learning Patterns . . . . .	83
6.3	Entity Similarity . . . . .	85
6.3.1	Proposed Framework Overview . . . . .	85
6.3.2	Generating Embeddings . . . . .	86
6.3.3	Information Source and Presentation Style . . . . .	86
6.3.4	Inferring Similarity . . . . .	86
<b>7</b>	<b>Experiments and Results</b>	<b>88</b>
7.1	Experimental Plan . . . . .	88

7.2	Experimental Settings . . . . .	89
7.2.1	Parameters and Hyperparameters . . . . .	89
7.3	Quantitative Evaluation . . . . .	90
7.3.1	Survey Overview . . . . .	90
7.3.2	Baselines . . . . .	92
7.3.3	Results . . . . .	95
7.3.4	Observations . . . . .	104
7.4	Qualitative Evaluation . . . . .	107
7.4.1	Qualitative Evaluation of ENSUS . . . . .	107
7.4.2	Inspecting the Shapley Values . . . . .	108
7.4.3	Visualization of Latent Dimensions . . . . .	109
7.5	Evaluating Scrutability . . . . .	112
7.5.1	Scrutability Results . . . . .	113
7.6	Performance Evaluation . . . . .	115
7.6.1	Performance Results . . . . .	115
7.7	Observations . . . . .	116
7.7.1	Presentation Style . . . . .	116
7.7.2	Descriptions and Justifications . . . . .	117
7.7.3	Explanation Efficiency . . . . .	117
<b>8</b>	<b>Discussion and Further Work</b>	<b>118</b>
8.1	Conclusion . . . . .	118
8.2	Further Work . . . . .	120
8.2.1	Finitetuning Hyperparameters and Model Architecture . . . . .	120
8.2.2	Self-Actualization . . . . .	120
8.2.3	Efficiency . . . . .	121
8.2.4	Improving ENSUS Architecture . . . . .	121
8.2.5	The Assumption that SHAP is Reliable . . . . .	121
	<b>Bibliography</b>	<b>122</b>





# List of Figures

2.1	Overview of recommender system methods from [87, 44]. . . . .	13
2.2	Perceptron from [30] . . . . .	22
2.3	The basic structure of a RNN. . . . .	24
2.4	NCF . . . . .	29
2.5	Neural News Recommendations with Personalized Attention (NPA) proposed in [99]. . . . .	31
2.6	Neural News Recommendation with Multi-Head Self-Attention [98]. . .	32
2.7	Deep Fusion Model (DFM) proposed in [55] . . . . .	32
2.8	Full architecture of DFM [55] . . . . .	33
3.1	The three orthogonal dimensions of explainable recommender systems illustrated as orthogonal vectors in a three-dimensional vector space. . .	40
3.2	Flame: Word cloud illustration of most important words for a user. Word size reflects the importance [101]. . . . .	51
3.3	Attention weights on user’s review text to discover important words [83]. Green color indicated high attention weight which indicates that the neural network consider the word important. . . . .	52
3.4	RippleNet: Illustration of how a Knowledge Graph (KG) can be used to model user preferences and provide explanations [97] . . . . .	53
3.5	The following chart provides a structured overview of relevant methods, means of visualization and evaluation in relation to explaining recommendations in a recommender system. . . . .	59
4.1	A visualization of feature importance using a sankey diagram, highlighting how certain features for a certain user contributed to the recommendation of the movie "Men in Black"[26] . . . . .	62
5.1	Histograms with key statistics of the Adressa dataset. . . . .	72
5.2	Histograms with key statistics of the MIND-small dataset. . . . .	75

6.1	Conceptualization of Explainable NewS recommendations Using Shapley values (ENSUS). . . . .	80
6.2	Overall architecture of ENSUS. . . . .	80
6.3	Architecture of the neural network for the proposed method . . . . .	81
6.4	SHARforce_plot [61] . . . . .	83
6.5	A high level architectural overview of the proposed justification by entity similarity, or relationship between read and recommended articles. As depicted . . . . .	85
7.1	Results on explanation through recommendation substantiation. . . . .	96
7.2	Results on explanation through highlighting the news category. . . . .	97
7.3	Results on explanation through conformity of news category and user profile, or <i>shared entity</i> [14] . . . . .	98
7.4	Results on explanation through abstract snippet[14] . . . . .	99
7.5	Results on explanation through entity relateness as proposed by Ripple-Net[97] . . . . .	100
7.6	Results on explanation through highlighting category and similarity to historic interactions. . . . .	101
7.7	Results on explanation through textual feature highlighting of Shapley values. . . . .	102
7.8	Qualitative results on explanation through visual feature highlighting of Shapley values. . . . .	103
7.9	Sankey diagram over the shapley values where the article itself is removed from the left siden. . . . .	107
7.10	The shapley values for top 5 recommendations for 30 randomly sampled users. . . . .	108
7.11	The shapley values for top 5 recommendations for 30 randomly sampled users where the maximum amount of articles in the click history is set to 10. . . . .	109
7.12	t-SNE visualization of embeddings from MIND article abstracts, colored according to article category . . . . .	110
7.13	t-SNE visualization of embeddings from MIND article abstracts, colored according to article subcategory . . . . .	111
7.14	$C(k, u)$ . . . . .	113



# List of Tables

1.1	Selected search terms and specific criteria related to the structured literature review. . . . .	6
2.1	Key characteristics of news articles as recommendable items. . . . .	10
2.2	Activation functions . . . . .	23
2.3	Table summarize activation function and loss function given problem type	23
3.1	Overview of information sources related to user preference and input[71]	42
3.2	Overview of information sources related to the decision inference process[71] . . . . .	43
3.3	Overview of information sources related to background and complementary information[71] . . . . .	44
3.4	Overview of information sources related to the alternatives and their features[71] . . . . .	45
3.5	Overview of recommendation inspired presentation styles of explanations[93] . . . . .	47
3.6	Evaluation criteria . . . . .	55
4.1	Evaluation criteria . . . . .	67
5.1	A comparison of available news datasets[100] . . . . .	71
5.2	Detailed statistics of the Adressa dataset . . . . .	72
5.3	Detailed statistics of the MIND-small dataset . . . . .	75
7.1	Parameters and hyperparameters . . . . .	89
7.2	Evaluation criteria or explanation goals as proposed by Tintarev et al.[93]	90
7.3	The seven evaluation statements with their corresponding evaluation goal.	91
7.4	Baseline explanations inspired by related work . . . . .	93

7.5	An overview of all explanations with respect to their type, information source, explanation model and presentation style. . . . .	94
7.6	Mean evaluation scores with respect to each explanation and evaluation goal. The highest score(s) for each respective goal is marked in bold. Here the Likert values are numbered, with 1 corresponding with <i>strongly disagree</i> and 5 corresponding with <i>strongly agree</i> . . . . .	104
7.7	Model fidelity at cosine similarity threshold of 0.6 . . . . .	111
7.8	Architectures used to evaluate the presence of the user profile. . . . .	112
7.9	The performance of different methods on the MIND dataset . . . . .	114
7.10	The performance of different methods on the Adressa dataset . . . . .	114
7.11	Left column report results in terms of Count@k with the original user profile. In the right "scrutinized" column, the user profile consists of randomly sampled topics. . . . .	114
7.12	The performance of different methods on the MIND dataset . . . . .	116
7.13	The performance of different methods on the Addressa dataset . . . . .	116



# Glossary

- AI** Artificial Intelligence. 33
- ANN** Artificial Neural Network. 20, 23
- BERT** Bidirectional Encoder Representations from Transformers. 18–20, 76, 109
- CBF** Content-Based Filtering. 1, 12–14, 16, 63, 71, 85
- CBOW** Continuous Bag-Of-Words. 18
- CBR** Case Based Reasoning. 46
- CF** Collaborative Filtering. 1, 5, 6, 11, 12, 14, 16, 46, 63, 65, 66, 85
- DL** Deep Learning. 1, 2, 20, 33, 38
- ENSUS** Explainable NewS recommendations Using Shapley values. x, 2, 3, 7, 78–80, 82, 85, 87, 88, 90, 105, 109, 113, 114, 116, 119, 121
- GDPR** General Data Protection Regulation. 65, 91
- HCI** Human Computer Interaction. 40, 41
- IR** Information Retrieval. 13
- KBF** Knowledge-Based Filtering. 12
- KG** Knowledge Graph. ix, 52, 53
- LIME** Local Interpretable Model-Agnostic Explanations. 34
- LRQ** Literature Review Questions. 5, 6
- LSTM** Long-Short Term Memory. 82
- ML** Machine Learning. 1, 2, 20, 33, 34, 38, 64
- MLP** Multi-Layer Perceptron. 20
- NLP** Natural Language Processing. 5, 17–19, 48, 54



**RQ** Research Questions. 2, 3, 88, 118

**SHAP** SHapley Additive exPlanation. 34, 82, 106

**XAI** Explainable Artificial Intelligence. 2, 4–6, 33, 38, 40, 41, 64



# Introduction

This chapter introduces the background and motivation for the thesis in section 1.1. Section 1.2 presents the overall goal and research questions. An overview of the initial research method and process is described in section 1.3 and 1.4 respectively. Lastly an overview of the thesis is presented in section 1.6.

## 1.1 Motivation

The last couple of decades have witnessed increased efforts in research and development on modern recommender systems. These increased efforts are mainly motivated by the promising efficiency and commercial value such systems provide in a digital society. Amazon<sup>1</sup> is usually credited to be among the first ones to embrace the potential of recommender systems to enhance user engagement in large-scale e-commerce platforms[14]. With recommender systems becoming increasingly more popular in other domains such as music, news, videos and more, the efforts on increasing their efficiency and accuracy have also increased.

In 2009, Netflix took their efforts to increase recommendation efficiency to new heights, introducing a 1.000.000 dollar award to the most efficient recommender implementation<sup>2</sup>.

The efficiency race have resulted in state-of-the-art recommendation techniques utilizing cutting edge approaches from Deep Learning (DL), introducing complex Machine Learning (ML) models that perform well beyond classical approaches such as Content-Based Filtering (CBF) and Collaborative Filtering (CF)[87]. Although the accuracy and commercial value of recommendations have increased with these efforts, a key requirement for the success and adoption of such systems is that users trust the system and its decisions. As the efficiency of ML based decision making models are increasingly embraced by system developers, the need for fair and transparent models which convey the reasoning behind their predictions have come of great importance.

---

<sup>1</sup>amazon.com

<sup>2</sup>[https://en.wikipedia.org/wiki/Netflix\\_Prize](https://en.wikipedia.org/wiki/Netflix_Prize)

This introduces the concept of *explainability* in recommender systems through providing explanations alongside recommendations, offering transparency and justifications for the recommendations. However, with the presence of modern ML algorithms, the explainability is further inhibited, resulting in the recommendation process being deemed a *black-box*, providing little to no leverage for transparency. The increasing concern related to *black-box* models have substantiated a whole new research area known as Explainable Artificial Intelligence (XAI), in which large efforts are laid in providing tools and approaches for increasing the transparency of ML based decision making systems.

The literature presents a number of approaches to explaining recommendations[39, 28, 12, 14], and existing work have demonstrated that explanations are beneficial for the success of explanations in a variety of ways, e.g. by helping users in making better and more informed decisions[93].

One area in which recommendation transparency is considered particularly important is *news*, both concerning the content and the technology used to expose citizens to relevant news. News readers increasingly consume content through personalized services that utilize recommender systems, as they aid users in alleviating the massive scale of available online news articles.

The following work is situated in a cross-section between the field of recommender systems, DL and XAI, and seeks to explore techniques for providing conspicuous explanations alongside recommendations of news articles.

## 1.2 Goals and Research Questions

This section introduces the goal and Research Questions (RQ)s of this thesis.

Explaining news recommendation is the goal *at-large* of the research presented in this thesis. However, due to the complexity with explaining ML methods directly, we wish to compare different approaches to explainability, namely how state-of-the-art descriptive methods from XAI compare to simpler justifications of explanations. For this reason, the overall goal of the thesis reads as follows:

**Goal** *Explore how state-of-the-art descriptive explanations compare to justifications in regards to providing trust, transparency and scrutability for a neural based news recommender system.*

Furthermore the work is split into three phases. The first phase is concerned with traversing the research landscape and related work within the field of explainable recommender systems. The purpose of this phase is to form a broad understanding of the current state-of-the-art in explaining recommendations, and leveraging crucial aspects of news related recommendation.

The second phase is concerned with designing and implementing a state-of-the-art explainable news recommender, resulting in a novel explainable recommender named ENSUS. ENSUS is based on the findings revealed in the research phase, and to evaluate the performance of the proposed methods in regards to transparency we compare it to a selection of baseline methods described in the literature.

### 1.2.1 Research Questions (RQ)s

The following RQ are explored and acts as a foundation for the thesis:

**RQ1** *What is explainability in the domain of recommender systems and what is the state of the art in providing explanations alongside recommendations?*

Research on explainability in recommender systems is still in its early stages, we want to research and understand the current landscape for how explainability is incorporated into recommender systems. Furthermore, we want to explore the options on how an explainable recommender system can be evaluated. To answer the question, we present a taxonomy of explanations in recommender systems.

**RQ2** *How does the explanations in the proposed method compare to the state-of-the-art explainable approaches?*

Based on concepts and approaches uncovered through answering RQ1 we develop a novel explainable recommender system, and we compare the performance with a selection of baseline approaches for explaining recommender systems.

**RQ3** *How does state-of-the-art descriptive explanations compare to justifications in terms of transparency and trust?*

As defined by [45], an explanation can be that of a description (concerned with revealing the actual mechanisms of recommender systems) or that of justifications (conveying a conceptual model that may differ from the underlying algorithm). Descriptions, or model concerned explanation methods are known for their complexity, but in return they are known to provide transparency to the otherwise complex decision making processes of neural networks. How does more novel, simplistic, justifications compare? Can we fully omit the black-box while still delivering transparency and trust? To answer this question we develop a justificatory model on top of the proposed explainable recommender system, that fully omit the black box. This is evaluated through comparing it to the descriptive explanations of the ENSUS model and other baseline approaches for explainability.

**RQ4** *What are the advantages and disadvantages of the proposed methods?*

This research question aims to discover and highlight the advantages and disadvantages of the proposed ENSUS model in a qualitative manner. By examining the Shapley values of the learned model we can tell whether the Shapley values can be used to explain the recommendations. Furthermore, experiments related to scrutability will tell whether it is possible to use user profiles to scrutinize recommendations.

## 1.3 Research Method

The overall ambition of defining and applying a research method is to propose and follow a detailed analytical process, that in turn will outline the deep knowledge and understanding of the state-of-the-art in explainable recommender systems.

The research and knowledge accumulated through the literature review was then used to formulate ambitions for the thesis, and substantiate the assumptions and methods proposed. In addition, an accumulation of research and knowledge on XAI in modern recommender systems was utilized answering RQ1 through constructing a detailed taxonomy on explainable recommender systems, depicted in chapter 3.

Moreover, an experimental plan was developed through insights on related work in evaluating explanations in recommender systems. With this knowledge, combined with a statistical and visual understanding of the recommenders efficiency a detailed evaluation-framework was developed. Finally, the contributions of this thesis were elaborated alongside future work, to further research the potential of descriptive and justificative explanation in news recommender systems.

## 1.4 Research Process

The research process for the thesis was divided into three distinct phases:

- Initial literature search for establishing a specific topic for the thesis
- Structured literature review protocol to find relevant literature for the thesis
- Structured literature review - implementing the review protocol

The individual phases are discussed in the sections below.

### 1.4.1 Initial Literature Search

The project description to this project was open and little restrictions was put upon the work. As a result, the goal for the initial literature search was to understand the state-of-the-art in recommendation systems, explainable recommender systems and XAI. To do so, search engines such as Google and Google Scholar were used. The main focus were to find and read surveys published over the last 5 years within the three topics. The initial literature search led to the surveys [102, 103, 36, 100, 78, 87]. On top of that the two most prominent recommender systems books were read: Recommender Systems by Charu C. Aggarwal [3] and Recommender Systems Handbook by Francesco Ricci, Lior Rokach, Bracha Shapira and Paul Kantor [82]. For XAI the main source of information was from the e-book Interpretable Machine Learning from C. Molnar [65]. It was discovered that there was a huge gap between the recommender books and the state-of-the-art in recommender systems; the recommender books focused on traditional algorithmic approaches to recommender systems while the state-of-the-art involved complex deep learning approaches utilizing techniques such as attention, recurrent neural

networks and convolution neural networks. For this reason, the initial idea was to experiment with attention networks to develop an interpretable recommender system for news recommendations. According to [102] attention models have eased the noninterpretable concerns of neural models. Furthermore, "the attention weights not only give insights about the inner workings of the model but are also able to provide explainable results to users" [102]. After some research it seemed too complex to further develop the state-of-the-art and the approach was thoroughly researched by much greater minds than ours.

The search shifted focus to experiment with Natural Language Processing (NLP) techniques and specifically looking at the embeddings learned by the neural network. NTNU has a large research community in NLP and it was therefore a natural approach to experiment with cutting edge, pre-trained transformer models for both English and Norwegian for representing the semantics of article content.

The search returned to the field of XAI and specifically model agnostic methods. The literature search discovered that LIME [81] was a major player in the XAI community and had gained massive interest over the past 5 years [65]. LIME is a surrogate explainable model that locally approximates the predicted output. However, LIME lacks the guarantee of accuracy and consistency [65] and can be impractical for industrial use as it is slow. For example, experiments performed by [26] shows that LIME required an average of 10-12 seconds to generate explanations for each recommendation in the experimental setup.

By PhD Robindra Prabhu at The Norwegian Labour and Welfare Administration (NAV) we were introduced to the concept of Shapley for explaining recommender systems. After meetings with Prabhu and Norsk Regnesentral it was decided to focus the research on using Shapley values to explain recommendations. The initial literature research discovered that Shapley values had received little focus in the explainable recommender systems community. A thorough review of the top 20 articles that emerge on Google Scholar when using the search words "Shapley Recommender Systems" shows that none of the resulting articles use Shapley values to explain the recommendations. As a result, it was decided to define the research objective as using Shapley values to provide explanations alongside recommendations.

## 1.4.2 Structured Literature Review Protocol

The review protocol functions as a framework for gathering relevant literature. The protocol contains specific guidelines for identifying and screening relevant literature and research to support the thesis, as well as suggested methods and criteria to ensure a sustainable research process. The protocol also reduces bias in the review process.

Two Literature Review Questions (LRQ) were defined to control the review scope, to narrow the initial spacious scope of the thesis. The initial scope of the thesis was restricted to *Explaining News Recommendations* with the Adressa dataset by [31]. Over time, this was narrowed down to providing explanations alongside recommender systems specifically implementing CF with multi-layer classification, using Shapley values for the explanatory parts. With this in mind, the following LRQs were defined:

- **LRQ1:** What information proves beneficial to gather during the literature search,

<b>Keywords</b>	Natural language processing, machine learning, data mining, user modeling, case-based reasoning, similarity-modeling and constraint satisfaction.
<b>Search Terms</b>	News recommender systems, explainable recommender systems, Collaborative Filtering (CF), XAI,
<b>Qualifying Criteria</b>	<ul style="list-style-type: none"> <li>• Literature should be related to Recommender Systems within the field of Artificial Intelligence.</li> <li>• The article seems relevant based on its abstract and conclusion.</li> <li>• Article has been cited in further work or similar research.</li> </ul>
<b>Evaluation Criteria</b>	<ul style="list-style-type: none"> <li>• Techniques and models used in the research should be reproducible.</li> <li>• Datasets and models used in the research should be open source.</li> <li>• The author(s) justify their design choices.</li> </ul>
<b>Inclusion Criteria</b>	<ul style="list-style-type: none"> <li>• The author(s) other works should display deep knowledge and experience in the field.</li> <li>• Work supporting underlying techniques and models should be dated past the year 2000.</li> <li>• Related work in the field of explainable recommender systems should be dated past the year 2010.</li> <li>• The studies should be written in English.</li> </ul>

Table 1.1: Selected search terms and specific criteria related to the structured literature review.

and how should it be gathered?

- **LRQ2:** How should this information be utilized, and what should it be utilized for?

The LRQs are supported by a search strategy for assisting in locating relevant literature. The search engines Google Scholar and IEEEExplore were used. The findings were evaluated in accordance to some defined *Qualifying*, *Evaluation* and *Inclusion* criteria to reduce the findings. In addition, relevant keywords and search-terms were defined to support the search. Table 1.1 provides an overview of these terms in addition to specified criteria.

### 1.4.3 Structured Literature Review

The final step in the research process involved an in-depth literature review. As the thesis is split between the discipline of recommender systems and explainable artificial intelligence this review was split into two respective parts.

First and foremost, an effort to determine the state-of-the-art in recommending items based on textual content in a collaborative manner was made, and previous work especially related to recommending news articles based on clicks and read-time were mapped.



In addition, state-of-the-art methods for providing explanations alongside neural and deep neural classification models were assessed.

Subsequently, efforts in providing explanations in collaborative based recommender systems were gathered, and further evaluated through evaluating the explain-ability of their respective underlying classification models.

## 1.5 Results

The proposed methods were evaluated according to qualitative and quantitative measures proposed in the literature, as well as statistical tools that allow a visual and statistical interpretation of the information sources utilized by the explanations. The quantitative evaluation was performed through a user survey in which participants evaluated each explanation with respect to seven evaluation goals. The explanations generated by ENSUS showed superior results in regards to perceived transparency with 96% of respondents agreeing to that the textual explanation increased the transparency, of which 67% strongly agreed. The visual explanation performed somewhat worse, as was expected due to the demonstrated superiority of textual explanation models compared to visualizations[51].

Furthermore, the proposed model for explanation through justification of the similarity between recommended and recently viewed articles outperformed ENSUS in terms of effectiveness and persuasiveness, while the models tied in terms of efficiency and trust, demonstrating that users appreciate contextual information about their recommendation.

However, while qualitative experiments on the embeddings substantiate how embeddings generated from news article abstracts are representative for news articles, a qualitative evaluation of the accuracy of the justification model showed that only 25% of recommendations could be explained through article relatedness when the threshold of similarity is kept at 60%.

## 1.6 Thesis Overview

The structure of the thesis is as follows: chapter 2 introduces core concepts, and provides the theoretical background information required to understand the contributions of the thesis. Chapter 3 provides a structured taxonomy on explanations in recommender systems. Then, chapter 4 provides an overview of related work and the state-of-the-art is highlighted. Subsequently, chapter 5 provides documentation on the data-sets utilized for this thesis. Chapter 6 presents the proposed model and its underlying techniques - based on relevant research, the constraints and conditions of the data-set as well as related work. Furthermore an evaluation of the model and its results are presented in chapter 7. Lastly chapter 8 discusses the results gathered as well as contributions and possible further work.



# Background Theory

This chapter lays the theoretical foundations required to understand the contributions of this thesis. Section 2.1 provides an overview of the problem context and attributes related with recommending news articles. Furthermore, theory on primary principles of traditional recommender systems are presented in section 2.2, followed by an overview of document representations in section 2.3. Furthermore, theory and neural networks and neural recommender systems are depicted in section 2.4 and 2.5. Lastly, we introduce some core concepts in explainable artificial intelligence in section 2.6.

## 2.1 Recommending News Articles

As society becomes increasingly more reliant on digital, more and more news readers tend toward reading news online with on-demand access to a vast amount of articles from different publishers. According to a report by Pew Research Center Journalism in 2018<sup>1</sup>, roughly 93% of adults in the US tend to read news online, either on desktop or mobile.

The purpose of news recommender systems is to aid the user in navigating this vast space of news articles, relieving the information overload by suggesting relevant articles based on an assumption of the user interests and preferences. However; the purpose of such systems is not restricted to news articles alone. Consequently, recommender systems, in general can be defined more formally as the following:

**Definition 2.1.1.** *Recommender systems can be defined as programs that attempt to recommend the most suitable items (products or services) to particular users (individuals or businesses) by predicting a user's interest in an item based on related information about the items, the users and the interactions between items and users [15].*

---

<sup>1</sup><https://www.journalism.org/fact-sheet/digital-news>

### 2.1.1 Characteristics of News Articles

As mentioned in definition 2.1.1 the recommendations predict a users interests based on information about the items, the users and interactions between the two. The accuracy and effectiveness of a recommender algorithm is strongly correlated to how this information is interpreted. Consequently; an in depth understanding of the characteristics of the items, users, and interactions is essential in building an effective news recommender system.

Before reviewing the challenges related to news recommender systems we will highlight some major characteristics that distinguish news recommender systems from other application domains such as music, books, restaurants and such. These characteristics are acknowledged in recent surveys on news recommender systems in particular[46, 78]. Table 2.1 provides an overview of relevant characteristics relevant for this thesis.

<b>Consumption Time</b>	The consumption time of a news story is highly correlated with the length of the article in terms of words. The user engagement time for articles between 101-250 words is 43 seconds, and 60 seconds for articles between 251-999 words[64].
<b>Life-Span</b>	Compared to books and movies, news articles have a dramatically shorter shelf-life. The relevance of news articles can be as short as maybe minutes, hours or barely a few days[78].
<b>Sequential Consumption</b>	News are often consumed in a sequential manner, where the user might seek to be updated on different stories at a time. Instead of being recommended similar news stories a user might prefer to read up on different topics[75].
<b>Diversity</b>	Music and movie consumers often consume one genre or category at a time, and might occasionally switch genre based on mood or change of interest. However; diversity in the news domain is not only related to keeping users engaged, but is also highly related to the issue of selective exposure, and is furthermore a key principle for a democratic society[78]. Diversity in online news is posing a major challenge for news recommender systems. Challenges related to diversity is further discussed in section 2.1.2.4
<b>Consumption Behaviour</b>	News articles are often consumed anonymously, and most often without explicit user profiles. This issue is most often mitigated by considering implicit signals such as click behaviour, time spent on page and browsing patterns. However, these implicit signals may be wrongfully interpreted, as a sign of appreciation or interest. Long read time may be caused by fatigue or idle time[78].

Table 2.1: Key characteristics of news articles as recommendable items.

### 2.1.2 Challenges for Recommending News Articles

The choice of recommender approach can impose several challenges. Challenges such as a *cold start*, *sparse data* and *long tail* are widely recognized in the literature, especially concerning recommender systems involving CF.

Although many challenges are intrinsic with the choice of the underlying recommender paradigm, the nature of the recommendable items themselves can impose contemporary challenges. In news recommendations the effectiveness of recommendations are in many cases highly reliant on some key aspects such as freshness, recency and trends. Furthermore, news outlets rarely allow users to rate the articles, posing limits on the user modeling compared to systems where detailed, explicit ratings are given by users.

Studies addressing news recommendation challenges include Raza et al. [78], Gulla et al. [32], and Moreira [87].

In addition to addressing challenges popularly, the literature have reached consensus on a wide range of challenges specific to news recommender systems, including *timelessness*, *user modeling*, *diversity*[32, 87, 78]. In addition, Raza et al.[78] identified a third challenge related to quality control of news content. As the latter is concerned with news content gathered from multiple sources, this thesis will focus on the issue of timelessness and user modeling, as it is highly relevant for the characteristics of the datasets utilized in this thesis.

#### 2.1.2.1 The Cold Start

The *cold start* is amongst the most known challenges in modern recommender systems. A cold start is related to the sparsity of information available, which in some cases can inhibit a recommender system. With users, a cold start is typically most evident when a new (or *cold*) user is introduced to the recommender system, where the recommender system has little to no knowledge related to the users preferences.

A cold start in the context of items is related to how an item has received few ratings, in which recommender systems implementing CF are exposed[56].

#### 2.1.2.2 Timelessness

Recommender systems in general are highly concerned with the relevance of the recommendable items. Recommendations with low relevance have shown to decrease concession and trust in the system and have a repulsive effect on the users[3]. On the other hand, relevant recommendations have shown to promote user satisfaction and concurrency between the system and the user.

News articles also have short life-cycles. From the moment an article is published the relevance may decrease, compared to that of e.g. a movie recommender system[49].

### 2.1.2.3 User Modeling

Users preferences are traditionally modeled through explicit and implicit feedback. Explicit feedback is considered *quantifiable*, e.g. the rating given by users of Amazon.com where an item is ranked on a numeric scale. In digital media applications, ratings are not typically given explicitly. In the case of online news services, users rarely rank the articles they encounter. For this reason, *implicit* feedback often acts as a proxy for a users interest[78]. Implicit feedback include click history, reading time, search history and percentage of an article that is scrolled.

Although these implicit signals can be used for inferring a users interest, a news recommender system must consider a variation of aspects in user modeling such as anonymity, passive news consumption, idle time, change of preference and short term intents.

### 2.1.2.4 Diversity

The issue of diversity have proven increasingly more relevant to news recommender systems in particular[78, 46]. Personalized news recommender systems are inherently selective. As recommendations are given to users, users are likely to interact with the suggested content compared to traditional broadcast content [10]. This presents the issue of *selective exposure*, research is taken from Festingers theory on cognitive dissonance; how people are more likely to attend to information that is consistent with their attitude rather than attitude-dissonant[25].

Scholars have shown concerns with the proliferating effect that personalized news recommendations can have on the general public. Especially how the public opinion can be degraded by isolating people from challenging perspectives by introducing selective exposure in news.

Research on selective exposure have shown that people prefer to view information that proves their own perspective [27, 35, 89].

## 2.2 Recommendation Paradigms

Researchers and business managers alike have recognized the potential of recommender systems, and various recommender techniques have been proposed since the mid-1990s.

Although the supporting paradigm of every recommender system is highly influenced by the recommendable items as well as its domain, most systems can be classified into four main paradigms based on some shared characteristics: CBF, Collaborative Filtering (CF), Knowledge-Based Filtering (KBF) and hybrid approaches[15].

The most frequently used techniques for traditional recommender systems have long been CBF and CF. This is mainly because they are based on rating data, which is relatively easy to collect and for which there are many available datasets[93].

Despite the fact that these paradigms differ in their implementation, their goal is shared; recommend the most suitable  $item(s) i \in I$  for the particular  $user(s) u \in U$ .

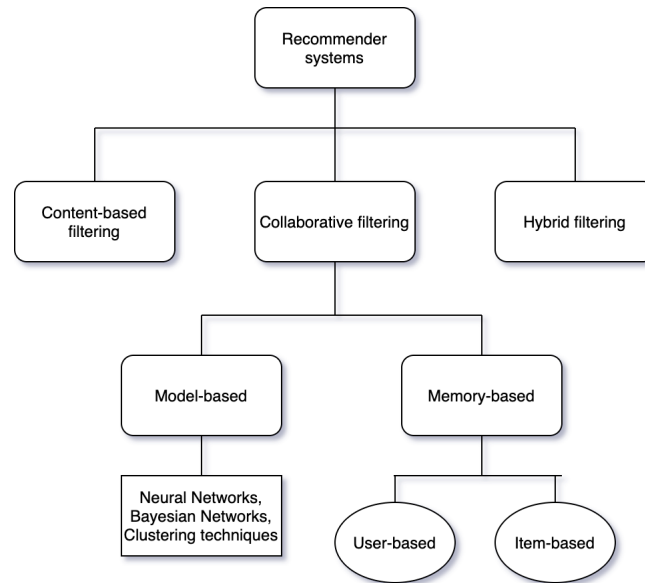


Figure 2.1: Overview of recommender system methods from [87, 44].

### 2.2.1 Content-Based Filtering

As the name suggests, content-based recommendation techniques utilize the *contents* of its recommendable items. This content varies according to the nature and characteristics of the item to be recommended. Nevertheless; when considering movies as with Netflix<sup>2</sup>, this content can be movie genre, actors, producers or length.

Consider a unique user  $u$ , that have ranked or viewed a subset  $i_u \in I$  of all available items  $I$ . CBF is performed by determining the similarity between this subset of recently liked items  $i_u$  and all available items  $I$  individually. Comparing raw text is cumbersome, therefore tangible feature vectors  $v_i$  of all recommendable items  $I$  are generated. By comparing the feature vectors  $v_{i_u} \in v_i$  based on items previously liked by a specific user  $u$ , to the feature vector of all available items, undiscovered items can be presented as recommendations based on their similarity to the ones already consumed by the user.

In the context of text corpora, such vector-representations – known as *embeddings* – can be based on basic term frequency as with TF-IDF[88, p.12] or more complex neural approaches such as with Word2Vec[29]. Determining the similarity between items can be performed through calculating the cross-product or cosine similarity between the respective embeddings. M

The similarity between such items is therefore restricted by the lexical meaning of the contents, consequently the semantic meanings are not included in the embeddings. Recent approaches suggested in

In the case of news articles, the recommendable attributes is mostly restricted to text documents. Hence, it is not surprising that many researchers rely on CBF techniques as text documents are easily analysed utilizing standard Information Retrieval (IR) techniques.

<sup>2</sup>netflix.com



Furthermore; an analysis of 112 papers in a recent survey on news recommender systems by Karimi et al.[46] show that 59 of the analyzed papers use CBF as the underlying paradigm.

## 2.2.2 Collaborative Filtering

In academic literature CF is the most common approach. The method — which in short is based on the "wisdom of the crowd" — is domain-independent in that it neither requires any knowledge about the domain nor the characteristics of the recommendable items themselves[78].

Since the Grouplens[49] project introduced CF on the Usenet news dataset in 1997, significant advances in collaborative filtering have been made. The recent decades have seen an increase in interest for such algorithms, presenting new concepts and models promoting the efficiency of recommendation algorithms.

In contrast to CBF, CF is not concerned with the contents or attributes of its recommendable items. Pure CF based recommender systems use correlations between users or items for projecting the *potential* interests of unseen items. In simple terms, the potential ratings of users are inferred through viewing what other users with similar interests have rated, thereby assuming in a "wisdom of the crowd" fashion whether or not a user would like a certain item.

### 2.2.2.1 Memory-Based Collaborative Filtering

Early implementations of collaborative filtering were based on the assumption that similar users like similar items, and would by example rate items equally. Such implementations utilize either user-user similarity or item-item similarity for projecting recommendations due to their approach of assuming interest based on the interests of *neighbours*. Such models are referred to as *memory-based*, as they utilize the entirety of the dataset for making predictions *upfront*, thereby requiring a lot of computer memory.

User-based CF utilize user profiles — that is, users and their previously rated items — by combined them as rows in a two-dimensional list, where all recommendable items are orthogonal to the respective users. This is known as an *user-item* matrix.

### 2.2.2.2 Model-Based Collaborative Filtering

In memory-based methods the prediction is specific to the instance being predicted. Such methods are often referred to as *instance-based learning methods*. In contrast, in model-based methods a summarized model is created up front. The learning phase is separated from the prediction phase, similar to what is done in traditional machine learning. Model-based methods rely on the fact that collaborative filtering is a matrix completion problem. Thus, a huge set of methods opens up. For example, the matrix completion problem is a generalization of the classification problem as it has a  $m \times n$  where the  $n-1$  columns are feature variables and the last  $n$ th column is the label. All entries in the first  $(n-1)$  columns are fully specified, whereas only a subset of the  $n$ th column is specified. The missing entries in the  $n$ th column have to be learned by the

model. This similarity between collaborative filtering and classification provides a richer set of possible methods to use when solving a recommendation problem. [3]

*Latent Factor Models* are a subgroup of model-based CF and is the prevalent technique in CF [3]. The idea behind latent factor models is that the preferences of a user can be modeled by a small number of latent factors by reducing the dimensionality of the original rating matrix. Latent factor models where we factor the rating matrix into one matrix for users and one for items is commonly referred to as matrix factorization models [3]. Note that in the following, we will assume that the rating matrix  $R$  have no missing entries as this is a valid assumption for our task at hand.

The  $m \times n$  rating matrix  $R$  is approximately factorized into an  $m \times k$  matrix  $U$  and an  $n \times k$  matrix  $V$ :

$$R \approx UV^T = \hat{R} \quad (2.1)$$

Where  $U$  and  $V$  are referred to as the user and item feature matrix respectively. The goal is to approximate the user and item feature matrix that minimize a loss function  $L(U, V|R)$ . To approximate  $UV^T$  to  $R$  we need to minimize the objective function  $J$ :

$$J = \frac{1}{2} \|R - UV^T\|^2 \quad (2.2)$$

where  $\|\cdot\|^2$  denote the squared Frobenius norm of the matrix. The smaller the objective function is, the better the quality of the factorization will be.

A row,  $\mathbf{u}_i$ , in  $U$  contains  $k$  entries and each entry in row  $i$  describes user  $i$ 's preference to one of the  $k$  concepts in  $R$ . Similarly, the  $j$ th row of  $V$  contains  $k$  entries and each entry represents the item's affinity towards one of the  $k$  concepts. The latent factors  $u_i = (u_{i1} \dots u_{ik})$  and  $v_j = (v_{j1} \dots v_{jk})$  are referred to as the *user factor* and *item factor*, respectively. The rating  $r_{ij}$  in  $R$  can be approximated by taking the dot product of the user factor and item factor:

$$r_{ij} \approx \mathbf{u}_i \cdot \mathbf{v}_j^T \quad (2.3)$$

Thus, equation 2.2 can be rewritten as:

$$J = \frac{1}{2} \sum_{i,j \in R} (r_{ij} - \sum_s^k u_{is} \cdot v_{js}^T)^2 \quad (2.4)$$

Each of the terms in  $(r_{ij} - \sum_s^k u_{is} \cdot v_{js}^T)^2$  is the squared error between the real rating  $r_{ij}$  and the predicted rating  $\hat{r}_{ij}$ .

The unknown variables  $u_i$  and  $v_j$  have to be learned. One approach in doing so is to use gradient descent (section 2.4.3) and updating the variables at each iteration:

$$\begin{aligned} u_{is} &\leftarrow u_{is} - \alpha \frac{\partial J}{\partial u_{is}} \\ v_{js} &\leftarrow v_{js} - \alpha \frac{\partial J}{\partial v_{js}} \end{aligned}$$

where  $\alpha$  is a constant. The updates can be executed until the variables converges.

### 2.2.3 Hybrid Systems

Hybrid recommender models are in essence produced through combining the efforts of different recommender models. The most widely implemented hybrid approaches involve both CBF and CF in unison. Hybrid systems mitigate the shortcomings of some models by incorporating the strengths of others, where e.g. a solely based CF model will suffer from the well known issue of a *cold start*, incorporating CBF can mitigate this shortcoming by suggesting related items, regardless of the specific user model being sparse.

### 2.2.4 Knowledge-Based Filtering

As previously mentioned, classical CBF techniques are restricted by the lexical meaning of its contents. In contrast, a knowledge-aware is said to have functional knowledge about the user, in that they have knowledge about how a particular item meets a particular users needs. Simple commercial recommender models — as in the case of Google — may simply attempt to deduce useful knowledge from a query formulated by a particular user, furthermore recommending specific items based on these.

## 2.3 Document Representations

The fundamental part of any language-related classification task is *representation*. The choice of representation as well as how the raw data is transformed to that representation can have large impacts on the result. In recommender systems, representations are utilized for determining relationships such as similarity between the recommendable items. When the recommendable items are news articles, representations are utilized for representing the contents of the news articles, substantiating the utilization of a variety of classification algorithms.

A particular text we choose to study is produced by one or more specific speakers or writers, in a specific dialect of a specific language, at a specific time, in a specific place, for a specific function[88, p. 13]. These variations, along with the length of the document and its vocabulary can differ greatly between documents. Therefore, representations of such documents are generated in advance, ensuring a fair comparison in a reproducible environment.

The choice of recommender paradigm and its underlying algorithms such as classifiers can impose restrictions on the representations, as some algorithms require a pre-defined type of input. Where decision trees allow almost any kind of input — be that discrete, continuous or canonical values, neural networks are restricted to vectors or normalized values often on a predefined range such as  $v = [-1, 1]$ .

### 2.3.1 Traditional Word Representations

The bag-of-words model is a way of representing a document as if it were simply a bag of words. The structure or order of words in the document is discarded, and the model

is simply concerned with the occurrence or frequency of words.

Some models — such as decision trees — might be able to interpret this format directly. However, most methods for classification and similarity will require this to be translated to a more normalized and tangible format in the form of values or vectors.

Assume the vocabulary of a given corpora is known in the form of a vector  $\vec{V}$ . Then each document  $\vec{v}$  could be represented as a sparse vector  $\vec{v} \in \vec{V}$  of length  $|\vec{v}|$ . Each position is here representative for a specific word in the document, and its value is the frequency of that word respectively.

The bag-of-words approach will normally result in very large vectors. Many researches therefore implement different means of pre-processing for reducing the dimensionality of the resulting vectors. One such technique is known as *lemmatization*; the task of determining if two words have the same root, despite their surface differences. A similar approach is known as *word stemming*, a simpler version of lemmatization where simply the suffixes of words are removed [88, p. 3]. Additionally, some researchers choose to completely ignore a whole class of words known as *stop words*; very frequent words such as i.e. *the*, *it* and *a* that bring little context to the document when the structure and order of words is discarded. Removal of stop words can be performed by defining a top 10-100 vocabulary entries by frequency in the training set, or by using one of many predefined stop word lists available[88, p. 60].

### 2.3.1.1 One-Hot Encoding

One-hot encoding is a simple and widespread approach for representing categorical data. A one-hot vector is a vector that has one element equal to 1 while all other elements are set to zero, hence the name "One-Hot". The encoding is performed through mapping each label to a binary vector, when encoding multiple elements the result yields a two dimensional vector, or matrix. For NLP related tasks, the vector length  $|V|$  corresponds to the vocabulary where each vector is corresponding to that words index in the vocabulary while all other values in the vector are set to zero.

One-hot encoding is widely implemented for evaluating and classifying categorical data, and makes it fitting for e.g. convolutional neural networks.

### 2.3.2 Word Embeddings

When the vocabulary size  $|V|$  grows, both one-hot encoding and bag-of-words will result in highly dimensional and sparse matrices. For instance, when dealing with a vocabulary with 50,000 words, a single word would be represented by 49,999 zeros and a single 1.

These methods also treat documents in an unstructured manner and often relinquish and change words through lemmatization and stop-word removal. This results in both models being inherently restricted to the lexical meaning of the documents, as the context of the documents is lost in the normalization process.

### 2.3.2.1 Word2Vec

Word2Vec is a technique for computing vector representations of words proposed by Mikolov et al.[63]. The word2vec toolkit consists of two models, Skip Gram and Continuous Bag-Of-Words (CBOW). Where CBOW is based on the assumption that the meaning of a word can be learned from its context, and it optimizes the embeddings in a manner that enables it to predict a target word given its context words. On the other hand, Skip Gram learns embeddings that can predict the context given a target word.

### 2.3.2.2 Glove

Drawbacks of Skip Gram and CBOW is how they are neglecting global information. On the contrary, GLOBAL Vectors for word representation (Glove) can capture corpus statistics directly. Proposed by Pennington et al.[77], the model combines the advantages of two major families in the literature, namely global matrix factorization and local context window methods such as the Skip Gram model proposed by Mikolov et al.[63].

However, it should be noted that the Word2Vec or Glove embedding for a specific word remains the same regardless of context, where i.e. the word embedding for *sentence* will remain the same for different contexts such as "a set of words that is complete in itself" and "the punishment assigned to a defendant found guilty by a court". In contrast, *contextual* embeddings capture these relationships through learning continuous representations for each word in the document.

## 2.3.3 Pre-Trained Contextual Embeddings

The role of context is imperative when comparing documents, as words that occur in similar contexts tends to have similar meanings. This link between similarity in how words are distributed compared to the similarity in their intrinsic meaning is known as the *distributional hypothesis*[88, p. 96]. Contextual embeddings utilize the potential of this linguistic hypothesis by learning representations of the *meaning* of words, rather than the words themselves. These representations are known as **word embeddings**.

Such embeddings, also known as dense word vectors, represent each word as a dense vector in an  $n$ -dimensional space, where typically  $n \ll |V|$ . These embeddings are powerful tools for modeling the semantic relation to individual words.

Typically word embeddings model the distribution of words based on their surrounding words the training corpus, further summarizing these statistics in terms of low-dimensional vectors. The geometric distance between the individual vectors represent the semantic relatedness between the words; thus implying a similarity.

### 2.3.3.1 BERT

Bidirectional Encoder Representations from Transformers (BERT)[23] is considered the state-of-the-art approach for a variety of NLP related tasks such as question answering, natural language inference and translation[23].

In contrast to previous efforts in language modeling where text sequences were considered in a left-to-right, right-to-left or combined manner, BERT implements a bidirectional training of a *transformer*; a popular attention model in language modeling. The transformer model allows an understanding beyond the simple lexical meaning of the words, in addition to simple semantics considering nearest-neighbours of words — allowing it to capture semantics beyond that of previous embedding-models[88].

For this reason, BERT is considered the state-of-the-art in contextual word embeddings, supporting research in detection of fake news, hate speech, sentiment analysis and other areas that would otherwise require human inference. In order to utilize the strength of BERT, the embeddings must be pre-trained on large corpuses of high-quality texts. Luckily, the recent decades have seen an increase in available corpora due to an increased effort of storing books and news-papers digitally. This have allowed the construction of multiple pre-trained BERT embedding model in many different languages.

### 2.3.3.2 BERT Variants

Pre-trained word embeddings have proven to be invaluable for increasing performance in NLP tasks involving text classification. Several approaches and pre-trained models on the BERT architecture have been proposed since [23] in a wide range of languages.

In terms of available text-corpora, Norwegian is a low-resource language, especially in comparison to English. This is quite evident considering Norway has merely 5.5 million inhabitants<sup>3</sup>, compared with English being the *lingua franca* of the world.

In a unique project started in 2006, the National Library of Norway is aiming at digitizing and storing all content ever published in Norwegian, making it available to the public. This includes of 500.000 books and 2.000.000 news articles<sup>4</sup>.

Large, available and high-quality text corpora is imperative for training effective dynamic embedding models. As demonstrated in previous work, a balanced national corpora — albeit smaller — consistently outperform large web-based corpora in semantic similarity evaluation[53].

The availability of high-quality corpora in Norwegian, as well as increasing interests for multilingual NLP, have given birth to several transformer-based text classification models for Norwegian. In addition, research in multilingual transformer models have provided several pre-trained BERT-based embedding models. In addition, some proposed models are tailored certain classification tasks such as tweets, where e.g. Nguyen et.al.[69] propose BERTweet; a large-scale language model pretrained for English tweets.

### 2.3.4 Sentence Embeddings

While BERT[23] and RoBERTa[60] have set the bar for state-of-the-art performance on sentence-pair regression tasks like semantic textual similarity, identifying nearest neighbours or most similar pairs in a collection of 10,000 sentences causes a massive

---

<sup>3</sup>ssb

<sup>4</sup><https://www.zdnet.com/article/norways-petabyte-plan-store-everything-ever-published-in-a-1000-year-archive/>

computational overhead, as it requires that both sentences are fed into the network thus resulting in 50 million inference computations. In other means, the construction of BERT makes it unsuitable for semantic similarity search.

Reimers et al.[79] recently released Sentence-BERT; a modification of the BERT network using siamese and triplet networks that is able to derive semantically meaningful sentence embeddings. This allows for a variety of new tasks that were previously not applicable using BERT or RoBERTa, such as clustering, semantic search and large-scale semantic similarity comparison. The latter is especially relevant for semantic similarity comparison in a news recommender system, where semantic similarities between titles or abstracts can be determined through similarity measures like cosine-similarity or euclidean distances[79].

## 2.4 Artificial Neural Networks

Classification tasks are generally concerned with recognizing features or similarities between a given set of observations, further organizing them into more abstract groups based on pre-determined criteria. A variety of approaches for classification is proposed in the field of ML and statistical analysis.

The recent decades has seen an increase in the utilization and research on classification models based on ML.

Artificial Neural Network (ANN)s are a collection of ML techniques based on the concept of an artificial neuron, inspired by biological neurons found in e.g. the human brain. The term *network* refers to how these neurons — like in the human brain — are employed in an interconnected manner, allowing them to solve complex analytical and classification tasks. DL, and is generally concerned with ANNs that consist of two or more layers.

It consists of multiple layers and each layer is built up of multiple neurons. Each neuron take input from one or multiple other neurons and compute an output which is passed to another neuron. ANN have gained widespread acceptance in several applications such as text classification, computer vision and speech processing. The basic motivation behind using ANNs is to extract useful features from the original attributes that are most relevant for the task at hand. ANNs have showed great success in extracting nonlinear features and they are able to extract richer sets of features, compared to traditional methods such as PCA[92].

A feedforward network, also referred to as Multi-Layer Perceptron (MLP)s, are a simple form of ANNs. These networks are called **feedforward** because the information flows through the network from the input towards the output and no feedback from the output are fed back into the input itself. The goal of a MLP is to approximate some function  $f^*$  by learning the values of its parameters. Each layer of the network can be considered as a function mapping some input to an output. Thus, the function  $f^*$  can be written as  $f^* = f_3(f_2(f_1(x)))$  where  $f_1$  is the first layer and  $f_2$  is the second layer and so on. The functions  $f_1$ ,  $f_2$  and  $f_3$  are connected in chain and the chain structures forms the structure of the network. The number of layers between the first and the last layer determines the *depth* of the network. [30]

Such deep learning approaches to news recommender systems have also proven effective addressing the challenges of timelessness and user modeling mentioned in chapter 2.1.2.

### 2.4.1 Overview of Training a Neural Network

The training process begins with the network receiving inputs. The data propagates through the network and results in an output. A loss of the network can be calculated based on a loss function. The loss function measures how well the algorithm performs on a single training example and is the error between the calculated output and the target value. The loss express how far off the output is.

In general, the network learns by adjusting its weights and it does so by propagating backwards through the network. This is called backward propagation and the aim is to determine how much each weight contributes to the error. The weights are finally updated and a new pass with forward propagation is executed. In theory, the next forward propagation would result in a smaller loss. [30]

The training can be generalized in three steps:

1. Forward propagation: Data is fed into the network and propagates forwards until an output is produced in the last layer.
2. Calculate the error: A loss function is used the calculate the error between the predicted value and the target value.
3. Backward propagation: Propagate backwards through the network in order to update the weights.

Now that we have described neural networks on basic level, we will now describe the mathematical properties of neural networks and deep learning.

### 2.4.2 Forward Propagation

As mentioned in the aforementioned subsection, forward propagation refers to the calculation of intermediate variables in order from the input layer to the output layer. In each layer, consisting of one or multiple neurons, each input is scaled with a weight according to its importance. This is followed by computing a weighted sum. The sum is then run through an activation function. For a neural network with a single neural, the forward propagation can be described as follows:

$$z = \sum_i (w_i x_i) + b \quad (2.5)$$

where  $w \in \mathbb{R}^{h,d}$  is the weight parameter,  $x$  is the input and  $b$  is the bias. Then the intermediate variable  $z$  is run through an activation function  $\phi$  in order to obtain the prediction value:

$$\hat{y} = \phi(z) \quad (2.6)$$



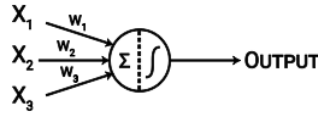


Figure 2.2: Perceptron from [30]

For a multilayer neural network, this can be generalized as follows: The weighted input of a node  $i$  in layer  $l$  receiving activations  $a$  from nodes in the previous layer  $l - 1$  can be described as [70]:

$$z_i^l = \sum_j^N W_{i,j}^l x_{i,j}^{l-1} + b_i^l \quad (2.7)$$

Before passing the value forward,  $z$  is passed through an activation:

$$a_i^l = \phi(z_i^l) \quad (2.8)$$

### 2.4.3 Learning with Gradient Descent

In order for the neural net to adjust its weights and biases, a common approach is to use gradient descent. Gradient descent is an optimization algorithm used to minimize the loss function  $J(w, b)$ . It can be understood like moving downhill in a landscape. The goal is to reach the bottom of the valley and the loss function forms the landscape. By taking a small step in the steepest direction, the parameters of the neural net is updated as follows:

$$w = w - \alpha \frac{\partial J(w, b)}{\partial w} \quad (2.9)$$

$$b = b - \alpha \frac{\partial J(w, b)}{\partial b} \quad (2.10)$$

where  $\alpha$  is step taken in each iteration and is commonly referred to as the learning rate.

Gradient descent provides an intuition of how the weights and biases are updated. However, the algorithm does not say how to calculate the derivatives. Backpropagation is a method to calculate the derivatives of all the nodes in the network. The mathematics behind backpropagation is rather complex and beyond the scope of this thesis.

### 2.4.4 Activation functions

The activation function decides whether the neuron should be activated or not and it also introduce non-linearity into the output of a neuron. Common activation functions are *Sigmoid*, *Hyperbolic Tangent*, *Softsign* and *Rectified Linear Units (ReLU)*. Table 2.2 lists the most common activation functions and their corresponding equation.

Name	Equation
Sigmoid	$\sigma(x) = \frac{1}{1+e^{-x}}$
Tanh	$\sigma(x) = \tanh(x)$
Softsign	$\sigma(x) = \frac{x}{1+ x }$
ReLU	$\sigma(x) = \max\{0, x\}$

Table 2.2: Activation functions

## 2.4.5 Output Functions

Output functions are identical to activation functions with the sole difference that they compute the output of the entire network. The choice of output function depends on the task at hand. For regression problems, a linear function or ReLU is a common choice. For binary classification, the Sigmoid function is used as the output of the Sigmoid function is a value between 0 and 1 which can infer how confident the model is of the example being in the class. The Softmax function is a generalization of the Sigmoid and works for problems with multiple output classes.

Type of Problem	Output Type	Final Activation Function	Loss Function
Regression	Numerical Value	Linear	Mean Squared Error
Binary Classification	0 or 1	Sigmoid	Binary Cross Entropy
Multiclassification	Multiple classes	Softmax	Cross Entropy

Table 2.3: Table summarize activation function and loss function given problem type

## 2.4.6 Topologies

Until now, we have discussed the ANN topology where the information flows through the network from input to output without any loops. However, other more complex topologies do exist such as Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN) and Long-Short-Term Memory Units (LSTM). In the following, we will present an overview of these three topologies. Deriving the mathematics behind these three topologies is a tedious and complex exercise. We will refer the reader to [30] and [70] for a complete explanation.

### 2.4.6.1 Convolutional Neural Networks

CNNs are a specialized neural network for processing multi-dimensional data and has proved to be a successful tool in deep learning [30]. Typically, a CNN have two components:

- Feature extraction part
- Classification part

In the feature extraction part, the network performs a series of *convolutions* and *pooling* operations. In terms of image classification, these two operations detect features

and filter out unimportant features. The classification part consists of traditional MLPs which assigns a probability to each image-instance to determine which class the image belongs to.

A layer in the feature extraction part typically consists of three stages. The first stage performs several convolutions to produce linear activations. A convolution is an operation on two set of functions  $f$  and  $g$ . The operation produces a third function  $f * g$  and this function express how the shape of  $f$  is modified by  $g$  and vica versa. The convolution operation is typically defined as:

$$s(t) = (x * w)(t) \quad (2.11)$$

where  $x$  is referred to as the *input* and  $w$  as the *kernel*. The output is typically referred to as the *feature map*. The purpose of doing convolution is to extract useful features from the input.

In the second stage, each activation is run through a nonlinear activation function, described in 2.4.4. In the third stage, a pooling function is used to modify the output of the layer further. For example, *max pooling* outputs the maximum output within a rectangular neighborhood. [30]

#### 2.4.6.2 Recurrent Neural Networks

Recurrent neural networks (RNNs) is a type of ANNs which uses sequential data, such as time series, natural language processing and speech recognition. RNNs are distinguished from the aforementioned typologies by their "memory". RNNs take information from prior inputs to influence the current input and output. This recurrence attempts to capture dependencies across time in the sequential data and RNNs assume that the current output depend on the former inputs and outputs.

The state of a RNN at time  $t$  is dependent on state of the network at time  $t - 1$ , the input  $x$  and a set of parameters  $\theta$ :

$$s_t = f(s_{t-1}, x_t, \theta) \quad (2.12)$$

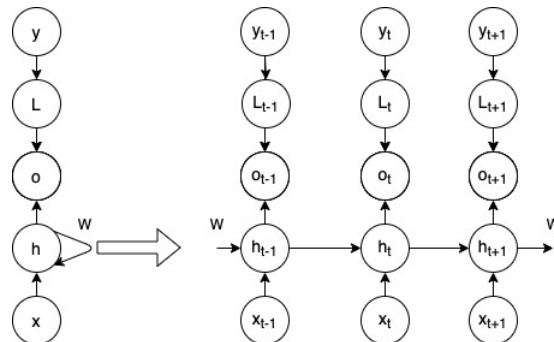


Figure 2.3: The basic structure of a RNN.

Figure 2.3 illustrates the basic structure of a RNN. The RNN maps an input  $x$  to an output  $o$ . The loss  $L$  measures how well the predicted output is to the target  $y$ .  $W$  denotes the weight matrix.

Using figure 2.3 we can develop the forward propagation equations. To simplify, we assume  $\tanh$  as activation function and  $\text{softmax}$  to obtain the predicted vector  $\hat{y}$ :

$$a_t = b + Wh_{t-1} + Ux_t \quad (2.13)$$

$$h_t = \tanh(a_t) \quad (2.14)$$

$$o_t = c + Vh_t \quad (2.15)$$

$$\hat{y}_t = \text{softmax}(o_t) \quad (2.16)$$

Two common variants of RNNs are Gated recurrent unit (GRU) and Long-Short-Term Memory (LSTM). The key difference between the two is that GRUs are faster to train as the GRU does not need memory units as in LSTM. [30, 70]

## 2.4.7 Regularization

A major challenge in learning deep neural networks is the complexity of the model. The complexity grows as we add hidden layers to the network. A common problem with complex networks is that they tend to overfit on the trained data. Overfitting happens when the model learns the details and noise in the training data, but it does not generalize on new, unseen data. As a result, the model may fail to predict future observations. Techniques that can help to reduce the complexity of the model are known as regularization techniques. Three regularization techniques are dropout, l2-regularization and batch normalization. [92, 70]

### Dropout

Dropout randomly deactivate nodes and is utilized during the training phase. At each iteration, we randomly select a fraction of nodes to be dropped from the network. Since the process results in multiple sub-networks, each neuron becomes not that reliant on neighboring neurons when learning patterns in the data. Since every sub-network has a different architecture, each node learns to be more agile to random modifications in the network architecture. This improves their generalization ability. [92]

### L2-regularization

L2-regularization is a technique used to force the model to discriminate weights with high values and thus reduce the complexity of the model. As a result, we now try to minimize both the loss and the complexity of the model. L2-regularization works by adding an extra term to the cost function and works as a penalty parameter. [70] defines the regularization term as follows:

$$\frac{\lambda}{2n} \sum_w w^2 \quad (2.17)$$

where  $w$  is the weights,  $\lambda$  is the regularization parameter and  $n$  is the size of the training set. For a quadratic loss function, adding l2-regularization results in the following equation:

$$C = \frac{1}{2n} \sum_x \|y - a^L\|^2 + \frac{\lambda}{2n} \sum_w w^2 \quad (2.18)$$

### Batch normalization

During training a neural network, the network typically takes a collection, or batch, of training examples as input. As the data propagates through the layers in the network it leads to an shift of the distribution in the data, also called a covariate shift. The consequence of a covariate shift is that the training and test data have a different distribution. Batch normalization ensure that the input to a layer is distributed around the same mean and standard deviation.

In addition to being a regularization technique, batch normalization also has the attractive quality that it can be used to speed up the training phase. It allows for much higher learning rates, which was the main contribution by the original authors who proposed batch normalization [43].

## 2.4.8 Attention mechanisms

Attention mechanisms have become a prominent technique in explainable neural recommender systems, usually to provide interpretable models over textual input features. Recent work such as [8, 59, 99] employ attention to identify the most important words in item descriptions or user reviews.

Attention was introduced in [5] in 2014 to overcome the bottleneck in encoder-decoder networks. The encoder process a input sequence, say a sequence of words, and compresses the information into a context vector  $c_i$ . The decoder transforms the context vector into an output. The bottleneck is the context vector since it is incapable of remembering long sequences. Attention was introduced to help memorize long sequences.

Consider an bidirectional RNN consisting of the hidden state  $h_i$  at timestep  $i$ . Using attention, the context vector is a sum of hidden states of the input sequence weighted by alignment scores:

$$c_i = \sum_{j=1}^{T_x} a_{ij} h_j \quad (2.19)$$

where  $a_{ij}$  is a number which tells the state  $i$  in the decoder how much it should pay attention to the state  $j$  in the encoder, e.g. the input word at position  $j$ :

$$a_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{T_x} \exp(e_{ik})} \quad (2.20)$$

$e_{ij}$  is a function learned by a small feed-forward neural network which takes the previous hidden state of the decoder,  $s_{i-1}$ , and the hidden state as input:

$$e_{ij} = \mathbf{v}^T \tanh(\mathbf{W}[s; h]) \quad (2.21)$$

where  $[s; h]$  is a concatenation of the two vectors and  $\mathbf{v}$  and  $\mathbf{W}$  are weight matrices.

## 2.5 Deep Learning in Recommender Systems

With today's hype of artificial intelligence and deep learning and the vast amount of research on deep recommender systems, it is appropriate the question the need for deep learning in recommender systems. In 2019 Zhang et al.[102] highlight the strengths of deep learning in recommendation systems; (1) *nonlinear transformation*, (2) *representation learning*, (3) *sequence modelling* and (4) *flexibility*.

Traditional methods such as matrix factorization and sparse linear model are linear models. Contrary to traditional linear models, neural networks are capable of modelling non-linearity in the data using activation functions such as Sigmoid, ReLU and tanh. The nonlinear transformation property makes it possible to capture complex user-item interactions[102]. Second, representation learning works by reducing high-dimensional data into lower dimensional data, thus reducing the complexity and making it easier to find patterns and discover anomalies. A large amount of data is generally available in real world applications. Making use of this information in order to better capture the relations between a user and an item, results in better recommendations[102]. Third, many recommendation systems are naturally sequential in the sense that a user view items sequentially in time. An user interests may change over time or items may become popular or unpopular over time. Both CNN and RNN is used to model such sequential tasks. Finally, deep learning techniques are flexible. With many deep learning frameworks such as Tensorflow, PyTorch and MXnet, it is easy and fast to implement, train and experiment with deep learning models.

Hidasi et al.[40] divide deep learning methods for recommender systems into five approaches: (1) *item embeddings and 2vec models*, (2) *extracting features from heterogeneous data* (3) *deep collaborative filtering*, (4) *autoencoders for CF* and (5) *session-based recommendations with RNNs*.

- (1) **Item Embeddings and 2vec Models** This direction is similar to latent factor models in the way that they model items as embeddings. Furthermore, they are used for item-to-item recommendations (CBF) without any user identification.
- (2) **Extracting Features from Heterogenous Data** Traditional hybrid recommender systems typically use one-hot-encoding, TF-IDF or LDA to represent contextual information. It is often easy to extract information from data using deep learning techniques. For example, CNNs can be used extract information from pictures and transformer models such as BERT can be used to represent words and sentences as n-dimensional vectors.
- (3) **Deep Collaborative Filtering** Deep Collaborative Filtering (DCF) is a general method for matrix factorization. By using neural embeddings to represent the user and the item respectively, the two embeddings can be merged/combined either by concatenation or a dot-product. To combat cold-start issues, one can put contextual information into the item embeddings and user embeddings.
- (4) **Autoencoders for CF** Autoencoders for CF is a subclass of DCF. Autoencoders are a special type of feedforward neural network where the input is the same as the

input. Autoencoders compress the input into a lower dimensional representation and then reconstruct the output from the representation. An autoencoder consists of an encoder and a decoder: the encoder compresses the input and produces the code, and then the decoder reconstructs the input using the code.

Autoencoders can be used in recommender systems by learning the structures of user-item-rating triplets and the learned representation can be used to predict users' ratings. However, this approach will result in lost information for news recommender systems as news content contains complex information [78]. Okura et al. 2017 [73] tried to solve this problem by adding noise in the input layer. This method is commonly referred to as denoising Autoencoder. In denoising autoencoders, the input is randomly corrupted and the autoencoder must then reconstruct the input, i.e. denoise. Instead of simply reconstructing the input, the model is forced to discover and learn robust features from the corrupted data.

- (5) **Session-based recommendations with RNNs** RNN model sequential data and is well suited to handle sequential user clicks and capture the dynamics of users' behaviors. GRUs and LSTMs have been used in news recommendations to capture the relationships between different clicks in a user history [87]. It is also possible to incorporate text such as titles and other side information into these models.

## 2.5.1 Methods for Neural Recommender Systems

This subsection presents relevant state-of-the-art neural recommender systems of relevance for this thesis.

### 2.5.1.1 Neural Collaborative Filtering

He et al. proposed in [37] a general framework named Neural network based Collaborative Filtering (NCF) which generalizes matrix factorization. The model consists of two input layers, two embedding layers and multiple *neural collaborative filtering layers* (commonly known as MLPs). The two input layers take the user feature vector and item feature vector respectively. The two feature vectors describe a user  $u$  and an item  $i$ . The embedding layers transform the user and item vectors from a sparse representation into a dense vector. The obtained user and item embeddings can be seen as the latent vector for the user and item. The embeddings are concatenated and fed into MLPs to map the latent vectors to a prediction score  $\hat{y}_{ui}$ .

For implicit feedback, the authors propose to minimize the objective function Binary Cross Entropy.

In addition to the proposed NCF, the authors extend the framework with a Generalized Matrix Factorization (GMF) component. The GMF component takes a user ID embedding vector and item ID vector and performs the element-wise product of the vectors:

$$\phi_1(p_u, q_i) = p_u \odot q_i \quad (2.22)$$

where  $p_u$  and  $q_i$  is the embedding vectors (latent vectors) of the user ID and item ID respectively.  $\odot$  denotes the element-wise product. Finally, the vector is sent forward through an activation function,  $a$ :

$$\phi_2 = a_{out}(\phi_1(p_u, q_i)) \quad (2.23)$$

The output of the GMF and NCF are concatenated and the combination of GMF and NCF are called NeuMF by the authors, which is short for *Neural Matrix Factorization*. I.e. NeuMF is an ensemble of GMF and NCF.

One major advantage of using the NeuMF framework is that the GMF component serves as a matrix factorization component and the NCF component can be used to add contextual information.

To evaluate the performance of the recommendations, the authors adopted *leave-one-out* evaluation. For each user, the latest interaction by a user is put into the test set. The remaining data is utilized for training. Experiments conducted by the authors conclude that NeuMF outperforms ALS on both Hit Ratio10 and NDCG10 on the Movielens<sup>5</sup>. Furthermore, the experiments show that NeuMF outperforms both NCF and GMF, and NCF performs better than GMF.

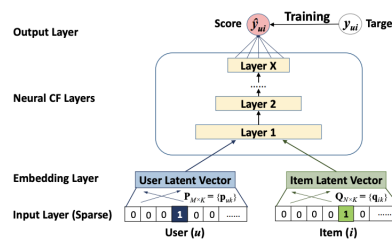


Figure 2.4: NCF

## 2.5.2 Deep Learning in News Recommendation Systems

As elaborated upon in section 2.1.2, recommending news articles introduces additional challenges such as decay of item relevance, extreme cold-start issues and the need for diversity in the recommendations. While traditional recommender systems typically represents users and items using IDs and their corresponding rating score, news recommenders are typically context rich to overcome the aforementioned challenges. This subsection presents an overview of the state-of-the-art in recommending news articles using deep learning methods. First, an overview of session-based news recommenders (using RNNs) are presented as they have become the prevalent technique to model dynamic user behaviors [87, 59]. Then DCF inspired news recommender approaches are presented which are more relevant for this thesis.

### 2.5.2.1 Session-based News Recommenders

Gabriel Moreira in [87] focus on session-based recommender systems for news in his thesis for the degree of Doctor of Science. He argues that session-based recommender

<sup>5</sup><https://grouplens.org/datasets/movielens/1m/>



systems are suitable to accommodate the challenges of news recommendations as they leverage information available in the current session. Additionally, session-based recommender systems usually deliver item-to-item recommendations in that they look at items that are similar in the current session. Moreira argues that this reduces the extreme cold-start issues and is suitable to recommend fresh news articles.

One example of a session-based news recommender is proposed in [72] by Okura et al. It is an embedding based method for news recommendation. They provide recommendations in three steps: (1) they encode articles using a denoising autoencoder, (2) generate user representations using a RNN with browsing history as input, and (3) rank articles based on inner product. For step 1, they feed the autoencoder a triplet consisting of three different articles where two of them have different but similar category and the third is from a dissimilar category. This is done to make sure the embeddings of similar articles are similar. In step 2, they do not randomly sample negative articles, but instead use the articles in which the user had not clicked on in the impression.

Other session-based news recommenders include [86, 52, 59] which are beyond the scope of this thesis.

#### 2.5.2.2 NPA

Wu et al. proposed NPA in [99]. NPA model the user and item representations in two separate towers - the news encoder and the user encoder . The news encoder use a CNN network to learn the representation of news articles based on titles. The user encoder learn user representations based on the user's click history. Attention mechanisms are used to score each word in the title in order to learn what words are important for the specific user. The user and item representations are merged by a *click predictor* using dot product and softmax in the final layer.

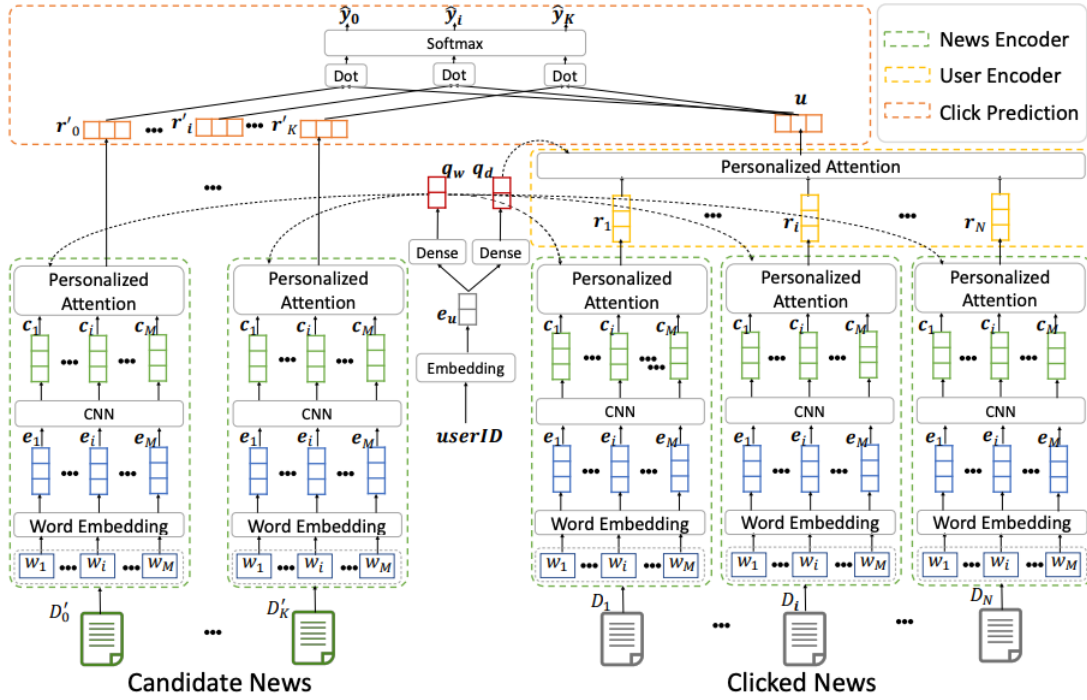


Figure 2.5: Neural News Recommendations with Personalized Attention (NPA) proposed in [99].

### 2.5.2.3 NRSM

Similar to NPA, Wu et al. model the news and user representations using a news encoder and a user encoder, in [98]. Different from NPA, the click predictor is a dot product between the browsed news, encoded by the user encoder and item encoder. Furthermore, the user encoder takes in multiple news articles which are individually encoded by the news encoder. NRSM uses self-attention to capture the relations between different words in the news title and the relations between different news articles in the click history. It also uses additive attention to learn what types of words in the news title are important to the user.

The authors use negative sampling to train the model. For each browsed article, they randomly sample  $K$  articles which are displayed in the same impression but not clicked by the user.

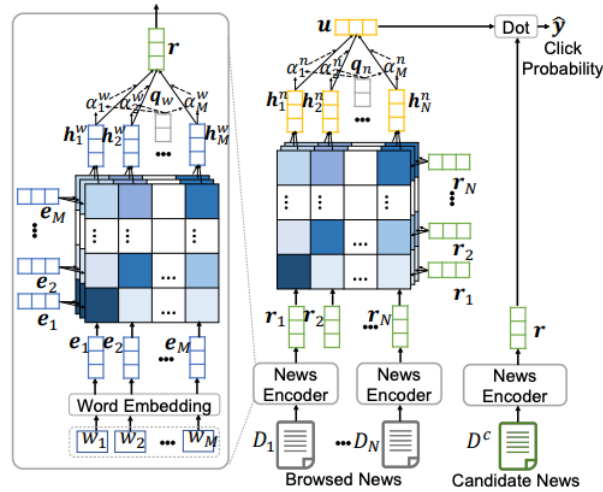


Figure 2.6: Neural News Recommendation with Multi-Head Self-Attention [98].

#### 2.5.2.4 DFM

DFM proposed by Lian et al. in [55] attempts to improve latent factor models using deep learning techniques by improving the user and item representations. They propose an *inception module* which learns item representations. The inception module is an extension of feed-forward. It learns multiple networks with different depths in parallel. The authors argue that different users have different distribution over the feature space. They solve this issue by learning different user representations using the inception module and fuse them via an attention network.

The training set consists of a user-article pair and a binary label  $y_i$  which denotes whether user  $u_i$  has read the news article  $v_i$ .

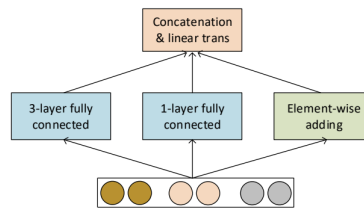


Figure 2.7: Deep Fusion Model (DFM) proposed in [55]

Figure 2.7 illustrates the inception module where different types of input features are fed different parallel networks. For example, categorical features are fed into the left and continuous features are fed in into the middle and left networks. Figure 2.8 illustrates the full architecture of DFM.

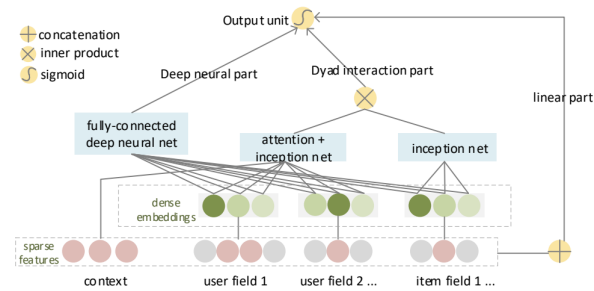


Figure 2.8: Full architecture of DFM [55]

## 2.6 Explainable Artificial Intelligence

While the very first Artificial Intelligence (AI) systems were easily interpretable, the last decades have witnessed the rise of DL based models, often comprised of hundreds of layers and millions of parameters. Such models have dramatically increased the complexity, further inhibiting interpretability and transparency compared to simpler models. AI methods incorporating such models are therefore deemed *black boxes* for their complex and intangible inner workings. *Transparency* can be viewed as the opposite of a black box, and the recent years have seen an increased effort in bringing transparency and explainability to such complex models.

The dramatic success in ML models has led to an explosion of AI based applications. However; the effectiveness of such applications will be limited to by the models inability to provide explanations tangible to humans.

Before core concepts and terminology can be established, a definition of the concept of XAI itself should be proposed. Although there is a lack of consensus related to defining XAI, Arrieta et al.[4] propose a broad definition based on the initial proposal by Gunning et al.[33], reflecting explicitly the dependence of an explainable model for an audience, as is appropriate for explainable recommender systems as well.

A broad definition of XAI could therefore read as follows:

Given an audience, an explainable artificial intelligence is one that produces details or reasons to make its functioning clear or easy to understand.

### 2.6.1 Methods for Explanation

XAI is generally concerned with explaining classification models, bridging the gap between end-users and the classification model through an appropriate interface.

Techniques in delivering transparency and explainability can be categorized to some degree. The literature makes a clear distinction among models that are interpretable by design and those that allow interpretation by certain XAI techniques. A widely accepted classification in this sense is that of *transparent-* and *post-hoc* explainability models, in which models are interpretable by design compared to models requiring an interpretability technique respectively.

## 2.6.2 Post-Hoc Explainability

Post-hoc explainability approximate the behaviours of a black-box by extrapolating relationships between feature values fed into the black-box model and the predictions or output of the model.

When discussing post-hoc approaches for explainability, the literature differs between *model-specific* (or *intrinsic*) and *model-agnostic* approaches. The great advantage of model-agnostic methods compared to model-specific ones is their flexibility in that they can be used on a variety of ML models, and is not concerned with the choice of model[4]. Since typically, multiple ML models are evaluated to solve a specific task, and when comparing models in terms of their interpretability, it is easier to work with model-agnostic methods, as the same method can be utilized for any model.

### 2.6.2.1 Model-Agnostic Methods

As the name suggest, model-agnostic techniques for post-hoc explainability are designed to adapt to any prediction or classification model, with the intent of extracting useful information about the models procedures — thereby *model-agnostic*.

In some cases, model-agnostic explainability techniques focus solely on explanation by simplification, generating proxies mimicking their antecedents with the intent of reducing the complexity of the model. Other model-agnostic approaches focus on extracting relevant information directly from the model, potentially presenting or visualizing this information, easing the interpretation of the model by reducing noise[4, 103].

### 2.6.2.2 Local Approximation

An approach to post-hoc model-agnostic explanations is determining the feature importance for a particular prediction through *local* approximation. Such explanations are closely related to local explanations, tackling explainability by segmenting the solution space and giving explanations to less complex solution subspaces that are relevant for the whole model[4]. Such feature relevance methods can be thought of as indirect methods for explanations.

Riberio et al.[81] propose a feature-based approach Local Interpretable Model-Agnostic Explanations (LIME) that fits a sparse linear model to approximate non-linear models locally.

SHapley Additive exPlanation (SHAP) is a framework for interpreting predictions and is used to explain individual predictions. SHAP leverages the idea of Shapley values for feature importance and was originally proposed by [84] in 1953. The approach is inspired by game theory, where input values are thought of as players, and where the feature importance in e.g. a neural network is determined through monitoring the players contributions to the output in a game theoretic environment.

Consider a housing price prediction problem. Using regression, the contribution to the final house price can be considered as a combination of the following boolean features: *is\_first\_floor*, *havebedroom* and *zipcode*. Furthermore, lets say the average predic-

tion is 200.000\$ for a given dataset and a given predictions yields 220.000\$. Our goal is to explain the difference between the actual prediction and the average prediction.

The answer could be: *is\_1st\_floor == True* contributed 15.000\$, *havebedroom == True* contributed 10.000\$ and *zipcode == 7010* contributed -5.000\$. The contributions add up to 20.000\$ which is 20.000\$ above the average prediction. Generally, the Shapley value is the "average marginal contribution of a feature value across all possible coalitions" [65]. This means that to compute the Shapley values we have to compute the prediction value for all possible combinations and coalitions of the three features. The Shapley value is the average of all the contributions to all possible coalitions. Continuing with the example, to determine the Shapley value for *is\_1st\_floor* the following coalitions are possible:

- No feature values
- *have\_bedroom*
- *zip\_code == 7010*
- *have\_bedroom + zip\_code == 7010*

For each of these coalitions we compute the predicted price with and without *is\_1st\_floor* and take the difference to get the marginal contribution. The Shapley value is the average of marginal contributions.

A major drawback of such models is that since the explanations are sourced from simpler surrogates, there is no guarantee that they are faithful to the original model. As with local explanations, they can be formed by means of techniques with the differentiating property that these only explain part of the whole system's functioning, and is not necessarily representative of the system as a whole[4].

### 2.6.3 Interpretable Models for Explainability

In contrast to model-agnostic explainability techniques, interpretable models achieve explainability by using a subset of algorithms that are interpretable on a modular level. Linear regression, decision trees and logistic regression are commonly used interpretable models. Most interpretable models covered in the state-of-the-art are interpretable on a modular level.

Interpretable models allow for inspection of the model components directly. For example, by traversing the edges in a decision tree it is easy to understand the prediction results. For attention models one can inspect the attention-weights to see what the model is focusing on.

#### 2.6.3.1 Linear Regression

In linear regression, a prediction model projects the target as a weighted based on a set of observations or features. Such models have long been implemented in applied

mathematics, statistics and economics[67].

$$y = \beta_0 + \beta_1 X_1 + \dots + \beta_p X_p + \epsilon \quad (2.24)$$

The predicted outcome of an instance is a weighted sum of its  $p$  features. The betas ( $\beta_j$ ) represent the learned feature weights, while the epsilon ( $\epsilon$ ) is the error, i.e. the difference between the prediction and the actual outcome [65, ch. 4.1].

A variety of methods can be used for inferring the optimal weights, where i.e. the ordinary least squares methods is widely used for locating weights that minimize the squared difference between the predicted and actual outcomes[67].

Determining the feature importance in regression models allows for a more detailed understanding, further promoting the interpretability of the models. Feature importance can also be combined with visual explanation approaches such as weight or effect plots, further substantiating the models explainability[65, ch. 4.1.3].

### 2.6.3.2 Decision Trees

Linear and logistic regression models fall short where the relationships between features and outcomes are non-linear or when the features interact. Decision trees build classification models in the form of a tree structure, where subsets of the datasets are created according to certain outcomes. To predict the outcome in each leaf node, the average outcome of the subset of training data in this node is used.

Decision trees are ideal for capturing interactions between respective features present in a dataset, and the data is often grouped in distinct groups easier to interpret than a multi-dimensional hyper-plane as with regression models. However, they fall short in dealing with linear relationships between an input feature and outcomes. Such predictions has to be approximated by creating a step function, which is not considered efficient[65, ch. 4.4].

### 2.6.3.3 Naive Bayes

Naive Bayes is a powerful yet rather simple classification model. The classifier uses the Bayes' theorem of conditional probabilities seen in equation 2.25. It calculates the probabilities for each feature independently, equivalent to a strong assumption — hereby *naive* — of conditional independence between the features.

$$P(A|B) = \frac{P(A|B)P(A)}{P(B)} \quad (2.25)$$

### 2.6.3.4 K-Nearest Neighbours

The k-nearest neighbour method can be used for prediction and classification tasks. It uses the nearest neighbours of a data point for prediction, hence its name. This model differs from other interpretable models as it is an instance-based learning algorithm.

The method is also inherently local, and there is no global weights or structures, and is thereby lacking a global interpretability.





# Taxonomy of Explanations in Recommender Systems

With the increasing efforts in utilizing methods from ML and DL in commercial recommender systems, the recommendation process of modern recommender systems can be viewed as *black boxes*; providing little to no transparency to the recommendation process.

As recommendations become more aware of user intent, and user information is being extensively collected and monitored, a lack of transparency to the reasoning behind such recommendations can affect users perceived degree of perceived intrusiveness and trust in the recommendation process [93].

The objective of this taxonomy is to define core terminology and aspects related to explanations in recommender systems, and further provide a hierarchical framework to which the different components of explainable recommender system research can be categorized.

As covered in a recent survey on XAI [4], the recent years have seen a dramatic increase in publications whose title, abstract and/or keywords refer to XAI. Inevitably, great efforts are aimed at defining and categorizing the concepts and convictions of XAI in both a general sense, in addition to that of an explainable recommender system. Consequently, the scope of the taxonomy will be restricted to concepts applicable in creating explainable recommender systems.

First and foremost, section 3.1 establishes core terminology to refrain from interchangeable misuse of concepts. Furthermore, section 3.2 provides an overview of the taxonomy and its dimensions. Following the overview, section 3.3, 3.4 and 3.5 discuss the taxonomy dimensions of *information source*, *presentation style* and *explainable method* respectively. Section 3.6 provides a presentation of metrics of evaluation. Lastly section 3.7 provides a summary of chapter and a tree-wise visualization of the taxonomy.

## 3.1 Establishing Terminology

Interchangeable use of concepts and terminology are prone to cause misunderstandings. Before further research can be discussed, an establishment of common ground is required. This is especially due to an interchangeable misuse of *explainability* and *interpretability* in the literature[4]. To avoid further misconceptions in the terminology, we define the following concepts in relation to explainable models and explanations:

- **Understandability**  
Denotes the characteristic of a model to make its functions understandable to humans, without the need for explaining internal structure and components.
- **Interpretability**  
Is defined as the ability to explain or provide the meaning in an understandable way to humans.
- **Explainability**  
An active characteristic of an AI model. It refers to the notion of explanation as an interface between humans and decision makers that is, at the same time an accurate proxy of the decision makers, and comprehensible to humans.

Furthermore, an explanation can be that of a *description* or a *justification*[45].

- **Descriptions**  
Descriptive explanations reveal the actual mechanisms that generate the recommendations, and is thereby mainly concerned with explaining the underlying recommendation algorithm.
- **Justifications**  
On the other hand, justifications convey a conceptual model that may differ from the underlying algorithm. For example, a book recommender system may be using an item-based k-nearest-neighbour algorithm to recommend books, but may justify a recommendation based on the fact that the book was written by the users favorite author.

While descriptions provide more transparency to the recommendation process compared to justifications, there are several reasons to why justifications might be preferred. Firstly, the underlying recommender algorithm may be too complex or un-intuitive to be simplified or described in more understandable terms. Secondly, developers might want to keep the underlying algorithm hidden due to commercial or competitive reasons. Lastly, justifications offer greater freedom in designing explanations, as they are not constrained by the recommender algorithm[45].

## 3.2 Taxonomy Overview

In this section, we provide a taxonomy of existing methods for explaining recommender systems. The taxonomy will function as a navigation tool in traversing the vast space of research in regards to explaining recommender systems. The taxonomy will also aid the readers in understanding the deduction and implementation of our proposed methods for explainability in a news recommender system.

The goal of explanations in recommender systems is to relate the recommended item to the intentions and interests of the user. As described in chapter 2.6, common approaches for establishing such a relationship is accomplished through the implementation of an intermediary entity that relates to both the user and the recommended item, thus functioning as an interface between the user and the recommender system; providing transparency and justification to the recommendation process.

In other words, implementations such intermediary entities are presented with three related challenges, what information should be utilized in the explanation, how should it be presented to the user, and how can it be collected.

For this reason, we classify existing research in explainable recommender systems with respect to three orthogonal dimensions visualized in figure 3.1, namely the *information source* (what knowledge or information is being used), the *presentation style* (how is this information or knowledge presented to the user) and lastly the *explanation model* (how is this information or knowledge gathered).

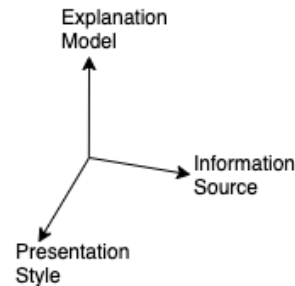


Figure 3.1: The three orthogonal dimensions of explainable recommender systems illustrated as orthogonal vectors in a three-dimensional vector space.

- **Information Source**

The first dimension is concerned with the *information source* that upholds the explanation.

- **Presentation Style**

The second dimension is concerned with the *presentation style*, representing the Human Computer Interaction (HCI) perspective of explainable recommender systems.

- **Explainable Model**

The third dimension is concerned with the *explainable model* that aims to provide explainability and interpretability in the recommender system, representing the XAI perspective of explainable recommender systems.

In contrast to the recent taxonomy proposed by Zhang et al.[103], we choose to include information source and presentation style in two separate dimensions. Zhang et al. argues that these dimensions are closely related because the type of information usually determine how the explanations can be displayed. However, this does not necessarily account for hybrid explanation implementations such as with the more recent work by Kouki et al.[51].

To allow for freedom in developing and designing explanations the taxonomy is structured in a *utility first* fashion, in that we focus on solid definitions on information sources instead of strict categories of presentation styles as e.g. performed by Zhang et al. Furthermore the taxonomy makes it easy to distinguish between the domains of data, HCI and XAI. However, we note that among the many possible classifications and taxonomies on explainable recommender systems, this is simply one approach that we think would be appropriate for the time being.

### 3.3 Information Sources

The first dimension is concerned with the information source of explainable recommender systems. Recommendation explanations can be generated from a variety of information sources. These sources can be known characteristics of items or users, or they can be more complex relationships such as the feature relevancy in a decision making process. Nevertheless, the kind of information that is used in an explainable recommender greatly affect the explanation, and combined with the *presentation style* makes up the HCI aspect of explainable recommender systems.

Nunes et al.[71] provides a detailed overview of sources of information that can be utilized in providing explanations in decision support and recommender systems. The information sources are organized in four main groups, namely *user preferences and input*, *decision inference process*, *background and complementary information* and *alternatives and their features*. Each group of information sources and their categories are elaborated upon in the following subsections.

### 3.3.1 User Preference and User Input

Explanations can be generated through providing users with information related to the provided user input and engagement in the system. The explanation can for example indicate which of the user preferences that were fulfilled with the recommended item and which that were not, or to what extent the system believes that the recommended alternative is appropriate given the stated or assumed preferences, or through highlighting which inputs that were the most decisive in determining the suggestion[71]. Table 3.1 provides an overview of information sources related to user preference and input.

User Preference and Input	Description
Decisive input values	Indication of the inputs that determined the resulting advice.
Preference match	Provision of information about which of the user preferences and constraints that are fulfilled by the recommendation.
Feature importance analysis	Describing the decision making process in terms of the relative importance of features, e.g. by displaying how a change of feature weights can affect the outcome.
Sustainability estimate	Indication of how the system believes that the user would evaluate the suggested recommendation, e.g. by showing a predicted rating.

Table 3.1: Overview of information sources related to user preference and input[71]

### 3.3.2 Decision Inference Process

The most common approach in the predecessor of recommender systems, *expert systems*, was to provide information about the inference process of a specific decision, e.g. in the form of traces. Table 3.2 provides an overview of information sources related to the decision inference process.

Decision Inference Process	Description
Inference trace	Provision of details of the reasoning steps that led to the recommendation, e.g. a chain of triggered inference rules.
Inference and domain knowledge	Provision of information about the decision domain or process, e.g. about the main logic of the inference algorithm, which for example can be presented as "We suggest this item because similar users liked it.
Decision method side-outcomes	Provision of algorithm-specific outcomes of the internal inference process, e.g. a calculated number that expresses the systems confidence.
Self-reflective statistics	Provision of facts regarding the systems performance, e.g. by informing the user how many times the system made decision suggestions in the past that were accepted.

Table 3.2: Overview of information sources related to the decision inference process[71]

### 3.3.3 Background and Complementary Information

Explanations can justify the decision making process through providing more information about the knowledge sources that are utilized or the relationship between entities that might not be apparent for the user. Table 3.3 provides an overview of information sources related to background and complementary information.

Background and Information	Description
Knowledge about peers	Provision of information about the preferences of similar users.
Knowledge about similar alternatives	Indication of similar alternatives that were an appropriate decision in a similar context in the past, made by the user or the system. E.g. items that the user or similar users showed interest in.
Relationship between knowledge objects	Provision of information about the relationship between features, or features and users.
Background data	Provision of (external) background data specific to the current problem instance, e.g. data derived from processing posts in a social network, which were considered in the recommendation process.
Knowledge about the community	Provision of information that supports the decision based on the behaviour and preferences of a community, e.g. showing the general popularity of the proposed alternative.

Table 3.3: Overview of information sources related to background and complementary information[71]

As such background knowledge is mostly associated with the decision inference process, additional complementary information can be gathered from external sources such as user reviews for specific items. Such information is categorized as *opinion based*, and can be beneficial for generating textual explanations and sentiment analysis[103].



### 3.3.4 Alternatives and Their Features

Approaches commonly implemented in the literature involve the relationship between the alternatives and their features, where some explanations depict lists of features for each recommendation, while others refer to dominant relationships through e.g. a top-N recommendation list. However, most explanations in this category are based on the feature relevance in the recommendation process[71]. Table 3.4 provides an overview of information sources related to alternatives and their features.

Alternatives and Their Features	Description
Decisive features	Indication of the features of the alternative that are key to the recommendation.
Pros and cons	Indication of key positive and negative features of the alternative.
Feature based domination	Justification of a decision in terms of the dominance of one recommendation compared to others, e.g. through showing that alternative recommendation was not selected as it was dominated by another.
Irrelevant features	Indication of features that are irrelevant for the recommendation.

Table 3.4: Overview of information sources related to the alternatives and their features[71]

## 3.4 Presentation Styles

This section we focus on the second dimension of the three-dimensional taxonomy to explainable recommender systems, which is the presentation style, or display style of explainable recommendations.

### 3.4.1 Recommender Inspired Styles

Early work by Tintarev et al.[93] provides a simple taxonomy that describe explanations inspired by a particular recommender algorithm. For instance, consider a recommender system whose underlying algorithm is solely based on item-based CF. Since the recommendations are generated based on the interests of nearby neighbours, a simple explanation can be on the form "*customers who bought this item also bought...*", as commercially utilized by e.g. Amazon<sup>1</sup>. The authors categorizes this as a *collaborative-based explanation*, as it bears a strong similarity to how the recommendation could have been generated. Furthermore they provide a simple taxonomy for recommender inspired explanation styles, such as *collaborative-based style explanations*, *content-based style explanations*, *Case Based Reasoning (CBR) style explanations*, *knowledge and utility-based style explanations* and *demographic-based style explanations*.

Notwithstanding, the underlying algorithm of a recommender system will to a certain degree influence the types of explanations that can be generated, hence the categorization of the following presentation styles. Details on each respective style is briefly covered in table 3.5.

---

<sup>1</sup>amazon.com

Presentation Style		Description
Collaborative-Based Explanations	Style	The most well known commercial implementation of such explanations are by e-commerce platforms such as Amazon.com. Such implementations assume that the user is viewing an item they are already interested in, furthermore suggesting items that users with similar interests also purchased. Hence the explanation " <i>customers who bought this item also bought...</i> ".
Content-Based Style Explanation		In short, content-based recommendations are generated considering the similarity between items previously rated by the user and new undiscovered items. In a same manner, content-based style explanations are based on item properties, and relatedness between the recommended items and other items.
Case-Based Reasoning (CBR) Style Explanations		There are many examples of explanation styles inspired by case-based reasoning. In short they use similarity between earlier cases or items to justify the recommendation. Its performance however varies based on which underlying algorithm is utilized.
Knowledge and Utility-Based Style Explanations		For all knowledge and utility-based explanation styles the assumed input is a description or <i>assumption</i> of the users needs or interests as previously described. Furthermore, the explanation may be performed through presenting the user with the inference match of a recommended item, as to justify <i>why</i> the item is recommended.
Demographic-Based Explanations	Style	For demographic style explanations, the assumed input to the recommendation system is demographic information on the user. From this, the recommender system users with a similar demographic profile. In a similar fashion, recommendations can be justified through informing the user about this relationship in a similar manner to that of collaborative-based style explanations.

Table 3.5: Overview of recommendation inspired presentation styles of explanations[93]

### 3.4.1.1 Shortcomings

As pointed out by [74], categorizing explanation styles in a one-to-one correspondence to specific recommender paradigms is considered simplistic, inefficient, and provide little to no leverage in explaining state-of-the-art neural based recommender systems that does not necessarily fall within a specific category of classical recommender systems.

However, we choose to include this categorization in the taxonomy as it provides a perspective on the historic development of the relationship between explanations and recommender systems, and also functions as a simple framework that can be easily implemented in more simplistic recommender systems that utilize classical recommender techniques as their underlying algorithm.

### 3.4.2 Feature Explanations

Feature based presentation techniques are concerned with simply highlighting or presenting certain features related to the recommendation process, recommendable items or the users depending on the application scenario. Research involving feature highlighting mostly incorporate the features in other presentation styles such as sentences or visualizations, as the features are rarely presented by themselves[103].

### 3.4.3 Sentence Explanations

Sentence based explanation techniques are presented to users in a textual manner. Sentence explanations can be viewed as a supplementary measure to other explanation styles or information sources, where contextual relationships and relevant information is added to the explanation.

The most common approach for sentence explanation is that of *template-based* sentence explanation, where information sources such as features or relationships are consolidated with a pre-defined sentence such as "You might be interested in [*item*], since it includes [*feature*]"[103]. The appealing feature of template-based textual explanations is how they can be easily tailored to a specific explanation, and allow for much freedom for the developers. Its ease of implementation and utility as well as wide implementation in both commercial and academic explainable recommender systems substantiates how template based textual explanation are by many considered the de-facto standard in presentation styles[103].

A more complex approach to sentence based explanations is that of *generated* sentences, where different techniques from NLP are used for generating sentence explanations directly. An appealing feature compared to that of template based ones is how they can support sentiment analysis, in which explanations can be perceived in a more humanely, or "word of mouth" based manner, as opposed to the fixed, generalized language of template based sentence explanations.

However, many approaches for explanation generation are reliant on user reviews as their training corpus, thus introducing noise due to how many reviews are not necessarily explanations or justifications for a users purchase[103]

### 3.4.4 Visual Explanations

Visualizations can assist a user in elucidating the result and reasoning behind a decision making process. Previous work in explaining recommender systems partly or fully through visualization can be grouped into two main categories, namely that of *chart-based* and *image-based* explanations.

#### 3.4.4.1 Chart-Based Explanation

Chart-based explanations are explanations where relationships, features or other relevant information sources are visualized through graphical interfaces such as barcharts, piecharts, visual rankings or similar. Charts are known for providing a familiar and intuitive interface for information sources that can be difficult to interpret in their raw format, or where the key information is first revealed through conceptualizing the relationships between different information sources.

Early implementations of chart-based explanations by Herlocker et al.[39] showed an increase in user satisfaction, where among other things, rankings by nearest neighbours were depicted in a histogram and a bar-chart. More recent work include the usage of simple sankey diagrams for visualizing feature importance[26], and more complex interfaces combined of 10 individual charts[24].

#### 3.4.4.2 Image-Based Explanation

To leverage the powerful intuition of visual imagery, explainable recommender research in utilizing images of recommendable items have been proposed. Such visual presentations are mostly based on information from underlying attention mechanisms that highlight certain relevant aspects on the image, such as the collar of a shirt or the waistband of trousers in the case of fashion recommendations in [20, 58]. Or that of relevant details on movie posters as with [59].

Due to the strong coupling between research on image-based explanations and that of deep image processing techniques, research on image-based explanations are still in the early stages, but with the continuous advancement in image-processing techniques we expect that images will be better integrated into recommender systems for both performance and explanation[103].

### 3.4.5 Hybrid Explanations

The presentation styles presented in this section can largely be denoted as *single style*, in that they provide explanations through a single presentation. However, recent research have studied the effect of hybrid explanations, and how to best present explanations that involve more than one presentation style.

## 3.5 Methods for Explaining Recommendations

Explainable recommender systems refers to methods and models that aims at making the behavior and recommendations of the system understandable to humans. The explanations help to clarify why the items are recommended. Explainable recommendation research consider the explainability of either the (1) recommendation model which is called *model-intrinsic* or (2) the recommendation results which is called *model-agnostic* or *post-hoc*[103].

Model-intrinsic approaches are often referred to as interpretable models meaning that the aim is to develop models that are interpretable in nature. Examples of interpretable models are decision trees, rule mining and attention-based networks. For these models it is possible to inspect the model components directly in order to understand the recommendations. For example, by traversing the edges in a decision tree it is easy to understand the prediction results. For attention models one can inspect the attention-weights to see what the model is focusing on. Specific to recommendation systems, such models directly leads to explainability of the recommendation.

The other philosophy for explainable recommender systems focus on the explainability of the results. The recommendation model is considered as a black box and a separate model is developed to explain the recommendation results [103]. Consequently, the explanation is generated *post-hoc*.

Zhang et al.[103] splits the current explainable recommender systems research into six topics for model-intrinsic approaches; *matrix factorization models*, *topic modeling*, *graph-based models*, *deep learning*, *knowledge graph-based*, and *rule mining*.

### 3.5.1 Matrix Factorization Models

One problem with matrix factorization models is the user/item embeddings are latent. The topics which influence the users decisions are modeled to be a predefined number of factors. However, it is difficult to know the exact number of topics in the document corpus and to extract the meaning of each factor.

Due to the popularity of factorization models in recommendation systems research, numerous solutions have been proposed to provide explanations for factorization models. One solution is proposed by Abdollahi and Nasraoui . The proposed method generate recommendations where a recommended item is explained by: *many users similar to you purchased this item*. Abdollahi et al. acheive this by adding a regularization term (see 2.4.7) to the objective function. The regularization term forces the user and item latent vectors to be close to each other when neighborhood users have rated the same item as the original user. As a result, the model naturally selects items that are rated by neighboring users.

SULM by Bauman et al. [9] uses sentiment analysis of user reviews to learn user's sentiment on item features. For example, if a user writes "The food is great", then "food" is extracted as a positive feature. Furthermore, if the sentiment analysis discovers that a user likes the feature "gym", then it can recommend hotels that has a gym and use "gym" as explanation. The features and sentiments are integrated into a matrix



Figure 3.2: Flame: Word cloud illustration of most important words for a user. Word size reflects the importance [101].

factorization model to predict the ratings. In addition to item recommendations, it can provide feature recommendations for each item as explanations. For example, it can recommend a restaurant together with what meal to order on which the restaurant performs well.

### 3.5.2 Topic Modelling

Topic modelling is widely used in the literature and refers to extracting the contextual meaning from text and using the topics extracted as explanations [103]. Explanations can be derived by showing the topic words that have had the most significant influence for the recommendations with the use of for example word clouds or bar charts.

FLAME proposed by Wu and Ester[101] uses CF and sentiment mining provide explanations alongside recommendations. Given review texts and user ratings on the items, FLAME uses *aspect-based opinion mining* learn each user's sentiment towards each possible topic. The explanations can be presented as word clouds where the word size is proportional to its sentiment.

McAuley et al.[62] argues that "in order to predict whether a user will like Harry Potter, it helps to identify that the book is about wizards". The authors combine review text with the latent user/item factors to discover and extract topics on items. In addition, the model can detect topics a user likes. It links each dimension of the latent vectors with a dimension from Latent Dirichlet Allocation (LDA) by projecting the user's latent vector dimension into the LDA dimension. For each item  $i$  they learn a topic distribution  $\theta_i$  where  $\theta_i \in \delta^K$  and  $K$  is a hyperparameter which denotes the number of topics. The topic distribution is a probability (stochastic) vector that describes to which extent a topic is present in all of the reviews on item  $i$ . The authors link the rating parameters  $\gamma_i$  and review parameter  $\theta_i$  by the transformation:

$$\theta_{i,k} = \frac{\exp(k\gamma_{i,k})}{\sum_k \exp(k\gamma_{i,k})} \quad (3.1)$$

which is added as a regularizer term in the objective function.  $k$  is a parameter fit during training.

### 3.5.3 Graph-based models for explainable recommendations

Graph-based models attempt to model either the user-user or user-item relationships as a graph and derive information from the graph to provide explanations.

Heckel et al.[38] use *co-clusters* which is an approach to group users and items with similar patterns. User might have several preferences and items may satisfy one or more of these preferences, so users and items may belong to several co-clusters. Explanations are generated in a collaborative fashion: "Item A is recommended to user Y with confidence 0.83 because user Y has purchased items B, C, D. Users with similar purchase history also bought item D"[38].

### 3.5.4 Deep Learning for explainable recommendations

Deep Learning leverage a vast amount of opportunities to derive recommendations and explanations and the amount of research is overwhelming. For this reason, we will only focus on using attention mechanisms to leverage interpretability.

Attention mechanisms is usually used to see which words in a text the model consider as important [103]. The results of such models typically leverage interpretability as illustrated in figure 3.3. Seo et al.[83] model user preferences and item properties using CNNs on review text. The method gather words with high attention weights to show which part of the review is more important.

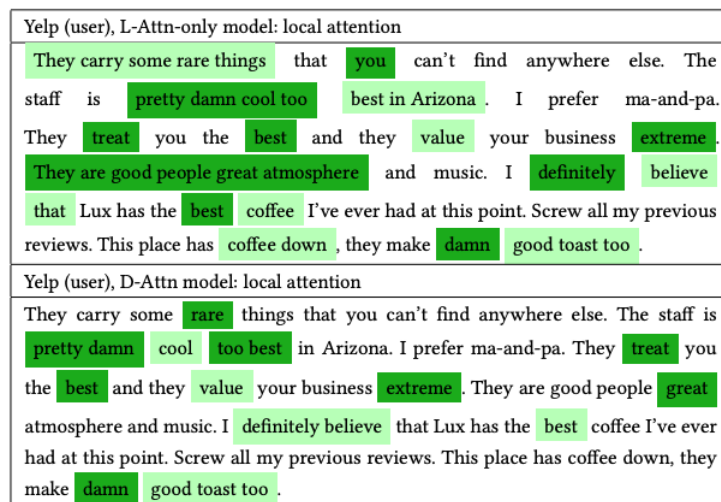


Figure 3.3: Attention weights on user's review text to discover important words [83]. Green color indicated high attention weight which indicates that the neural network consider the word important.

#### 3.5.4.1 Knowledge Graph-based explainable recommendations

KG is a domain of knowledge and it provides a structure and a relation for the data. Using KGs it is possible to extend a user's interest and deriving new entities.

RippleNet by Wang et al. [97] use a KG as side information the address the sparsity and



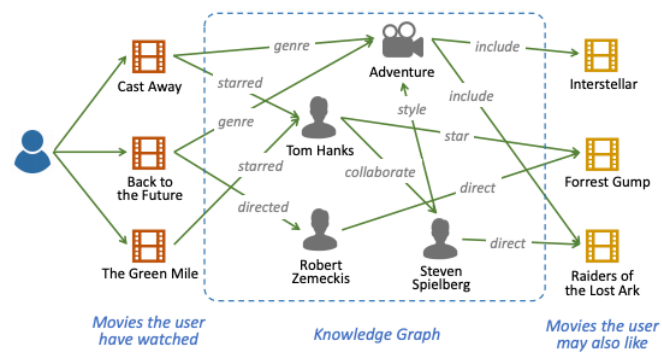


Figure 3.4: RippleNet: Illustration of how a KG can be used to model user preferences and provide explanations [97]

could start problems of CF. By using the KG, the algorithm "ripples" through links in the KG to extend user preferences by iterating over the user's click history. Explanations can be generated by traversing the edges in the graph from the user to the recommended item. Figure 3.4 illustrates how RippleNet utilize a KG to provide recommendations and explanations. By traversing the edges, the explanations can be generated. For instance: "We recommend Forrest Gump since you have watched Cast Away starring Tom Hanks and Tom Hanks also plays in Forrest Gump".

### 3.5.4.2 Rule Mining

Rule mining is one of the earliest approaches for leveraging explanations alongside recommendations. It is a popular method for generating explanations as it is easy to implement and can generate straightforward explanations [103].

Balog et al.[7] recently proposed a set-based technique for transparent, scrutable and explainable recommendations. They utilize a *user profile* which provide a textual description that summarizes the system's understanding of the user's preferences [7]. In this way, the user can scrutinize this summary and modify his user model. For example, a user summary can be summarized as follows:

- You like movies that are tagged as "action", especially those that are tagged as "violent", such as Aliens.
- You like movies that are tagged ad "cheesy", such as Who Framed Roger Rabbit?

### 3.5.5 Model Agnostic Methods

An alternative to the model intrinsic methods like those discussed above is model agnostic methods. Here, recommendations and explanations are generated using two different models. The model agnostic approach is common in neural recommender systems as such models are usually difficult to explain [103].

## 3.6 Evaluating Explanations

The state-of-the-art presents a variation of approaches to evaluation. For this reason, this section is concerned with roughly categorizing the means and metrics of evaluation. Implementations and approaches present in relevant research will be elaborated upon in the next chapter on related work, namely section 4.4.

### 3.6.1 Means of Evaluation

Next, we present an overview of evaluation types researchers applied to assess or compare different explanations provided by the system. This overview is based on an assessment of research depicted in the following research and surveys [103, 93, 71, 14].

#### 3.6.1.1 User Survey

User studies are the predominant approach for evaluating explanations in recommender systems. This is expected, as there is no *formal* definitions or benchmarks of what a perfect is. For this reason, the only way to evaluate the provided explanations is to capture the subjective perception of users.

#### 3.6.1.2 Online Evaluation

As user surveys are excellent for capturing a users opinion on explanations, they may affect the outcome in that the user is aware of the controlled environment, and might divert from how he/she would naturally respond to explanations on a day to day basis. Online methods of evaluation are also concerned with the subjective perception of users, but may be able to monitor the impact of the explanations in correlation with a commercialised product. However, due to the limited information that can be collected in an online environment, it is usually easier to evaluate *persuasiveness*, as to whether or not the explanations made users more likely to accept the recommendation.

#### 3.6.1.3 Offline Evaluation

Another mean of evaluation is that of a qualitative offline evaluation. Zhang et al.[103] mention two approaches, namely evaluating the percentage of explanations that can be generated, and that of measuring the quality of the explanation directly, requiring a benchmark tool, making it fitting for a NLP based evaluation of e.g. automatically generated textual explanations.

Furthermore, tools for offline evaluation will depend on the information sources, methods and presentation styles utilized in generating the explanations. And as Zhang et al. points out, more tools and frameworks for offline evaluation are yet to be proposed for evaluating explanations.

### 3.6.2 Metrics for Evaluation

Explanations can largely affect how people respond to recommendations. Explanations are provided for attracting more users and convincing existing users to embrace the recommendations, fueling the user modeling process allowing further exploration of unseen items. As argued by Bilic et al.[12], the most important contribution of explanations is not to convince users to adopt the recommendations themselves, but to allow them to make accurate and informed decisions about which recommendations they wish to utilize; thus focusing on user satisfaction rather than promotion of items.

Nevertheless, measuring the degree to which a user feels satisfied with an explanation is difficult, as they are often restricted to subjective interpretations.

First proposed in her PhD thesis on explainable recommendations and later published, Tintarev et al.[93] provides a *first of its kind* compilation of explanatory criteria in recommender systems, similar to those desired in early expert systems as demonstrated by Buchanan et.al.[16].

Table 3.6 summarizes previous evaluations of explanations in recommender systems and the criteria by which they have been evaluated. Although some of the criteria may interact, it is important to view them as distinct.

Evaluation criteria	Definition
Transparency	Explain how the system works
Scrutability	Allow users to tell the system it is wrong
Trust	Increase confidence in the system
Effectiveness	Help users make good decisions
Persuasiveness	Convince users to try or buy
Efficiency	Help users make decisions faster
Satisfaction	Increase the ease of usability or enjoyment

Table 3.6: Evaluation criteria

#### 3.6.2.1 Transparency

In the case of explanations, transparency help the user to understand how the recommendations were selected. Transparency also aims to help the user to understand how the recommendation fits their needs. Consider a movie recommendation systems always recommending for a user who actually likes action movies. Such a user should be given an explanation.

#### 3.6.2.2 Scrutability

Recommendation systems usually make assumptions about the user, the items or both. Explanations may help to correct misguided assumptions and such explanations is said

to increase the scrutability of the recommender system [93]. I.e., scrutability allow the user to tell the system that something is wrong. The dating app Tinder<sup>2</sup> is a great example of a systems which supports scrutability in the sense that it allows the user to "swipe" away potential recommended matches.

### 3.6.2.3 Trust

Trust is defined as the users' confidence in the system. Tintarev argue that trust is highly correlated with the accuracy of the recommendations. Poor recommendations yields high churn and vica versa. However, the author argue that explanations can, to some extend, compensate for poor recommendations and thus increase the trust of the system. A user may be more forgiving if they are giving an explanation containing the confidence score of the recommendation.

### 3.6.2.4 Persuasiveness

Persuasiveness is related to how well the system convince the user. In the case of explanations, persuasiveness can be measures in a number of ways. For example, it can be measured as the difference between two ratings: First, let the user rate the recommendations without explanations and then let the user rate the recommendations with explanations. Another possibility is to measure the change in click-through rate on recommended items.

### 3.6.2.5 Effectiveness

Effectiveness is related to how well the system help the user to make good decisions. As a result, effectiveness is dependent on the accuracy of the system since the user cannot make good decisions if the recommendations is not correct. Effectiveness can be evaluated by measuring the difference between a user rating before and after the user consuming the item. For example, a user can be asked to rate a news article after reading only the title and then again after reading the article. If the opinion did not change much, the system is considered effective. As a result, effectiveness is closely related to precision and recall [94].

### 3.6.2.6 Efficiency

Efficient explanations is related to how well the system help users to make decisions faster. Explanations made to increase efficiency will help the user to understand the difference and the relation between competing options. Efficiency can be measured by tracking the number of interactions with the system before finding a satisfying item.

---

<sup>2</sup><https://tinder.com/>

### 3.6.2.7 Satisfaction

Explanations that increase the acceptance of the systems as a whole are said to increase satisfaction. One way of measure the satisfaction is to conduct a user walk-through for finding a relevant item. In such a case, one can use quantitative measures such as the number of times a user found an irrelevant item or the ratio between irrelevant and relevant items found by the user. A way to measure the satisfaction more directly on the explanations is to ask the user if the system is enjoyable before introducing explanations and then again after presenting explanations.

### 3.6.3 Levels of Explanations

As mentioned, the explainable aspect of an explainable recommender system was defined as an interface between the recommender system and the human end user. This interaction is referred to as *human-machine-interaction* in the literature, and is largely concerned with how a computer representation is communicated to the end user.

In a recent study from 2019 on explaining user profiles from aggregated reading data in a content-based news recommender system, Sullivan et.al.[90] identified three *levels of explanation* that a user profile can serve. The authors propose an explainability framework for categorizing explanations in recommender systems in a hierarchical manner, where each level of the framework represent a certain function that a user profile can serve.

The levels are structured in a hierarchical manner, where each utilizes the information from the previous, steadily increasing transparency and ultimately self-actualization.

- **Level 1: Transparency**

The first level consists of the raw data the platform currently has on the user based on his or hers reading patterns and historic interactions. This level is necessary as it provides the foundations for the following. The transparency layer can also deliver insight for the user through descriptive information; simply visualizing or describing the raw data that is utilized for the recommendations, which can further assist the users in answering questions regarding their historic interactions and reading behaviour, through e.g. visualizing the distribution of monthly read topics.

- **Level 2: Contextualization**

The second layer combines the users historic interactions with news articles, contextualizing it with that of their community. This helps users understand how others are using the news platform, and can be exercised through side-by-side distribution of monthly readings. Such explanations have been shown to help users answer questions about how their news consumption habits compare to others[93].

- **Level 3: Self-Actualization**

The third layer foster self-actualization through supporting epistemic goals, allowing users direct control over which goals they wish to actualize. The user should be presented not only with the goal itself, but a short textual description should accompany the goal. The recommender system should support users in understanding their unique tastes and preferences [48].

## 3.7 Summary

Based on a thorough and systematic research process and literature review as described in section 1.4, we propose a condensed but far-reaching taxonomy that capture relevant facets that one might consider when developing explainable recommender systems.

Table 3.5 depicts the relevant aspects related to explaining recommendations in a recommender system.

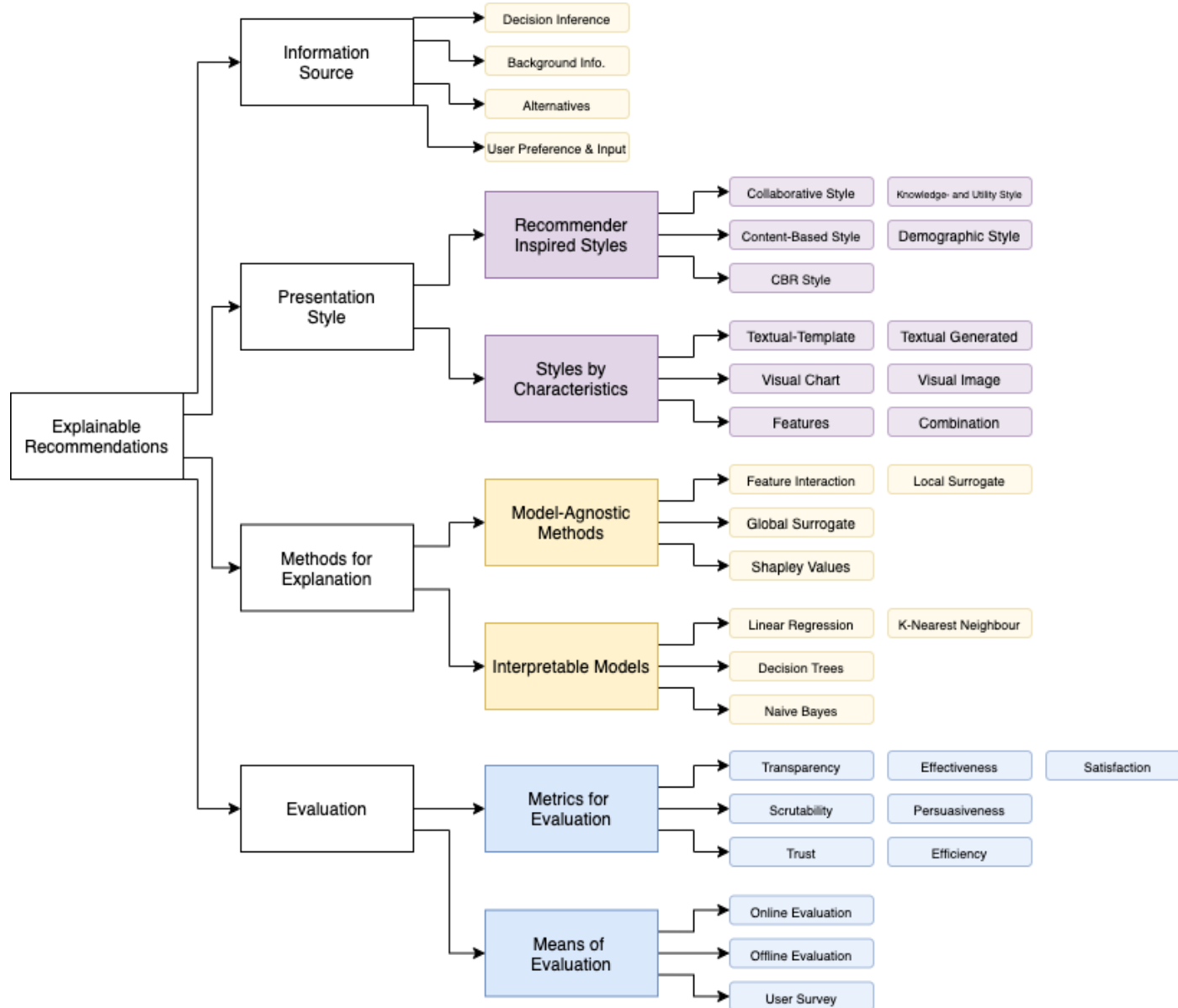


Figure 3.5: The following chart provides a structured overview of relevant methods, means of visualization and evaluation in relation to explaining recommendations in a recommender system.





# Related Work

This chapter provides an overview of related work on generating, presenting and evaluating explanations in recommender systems. The following chapter serves as a non-technical summary that motivated the remaining chapters while also functioning as a detailed addition to the taxonomy provided in chapter 3.

## 4.1 Explainable Recommendations

Based on the concepts and categories of explainable recommender systems introduced in the proposed taxonomy from chapter 3, the following sections are concerned with discussing the intersection of the taxonomy dimensions of information source and presentation style through commercial and academic implementations depicted in the literature.

In addition, we consider the aspects of evaluation to be paramount in respect to the contributions of this thesis. On that note, we have dedicated section 4.4 to related work, frameworks and tools for evaluating explanations in recommender systems.

### 4.1.1 Explainable News Recommendation

Although the chapter is called *related work* and the goal at-large for this thesis is to develop an explainable *news* recommender system, this chapter will not be entirely focused on related work in recommending news. The reasoning behind this is that there is limited research on explainable news recommendation. We suspect that this correlates with there being few available datasets tailored for news recommendation. For this reason, the following chapter will cover a variety of commercial and academic recommender systems, but as described in the literature review protocol from section 1.4.2, the research covered should be relevant or transferable to news recommendation.

## 4.2 Source and Presentation of Explanations

In short, the presentation styles of explanations are concerned with highlighting relevant aspects of the recommendation process. In addition to the early works on explanation styles by Herlocker et al.[39] and the more recent taxonomy of presentation styles presented by Zhang et al.[103], a variety of presentation styles have been implemented in commercial and academic recommender systems that facilitate explanations.

Many of these explanation styles are also directly comparable to the dimensions of information source and presentation style depicted in the taxonomy, and also the earlier taxonomy provided by Tintarev et al.[93] where e.g. the characteristics of content-based style explanations overlap with those of feature-based explanations. As pointed out by [93], the explanation style may simply follow the style of a recommender paradigm, irrespective of how the recommendation actually was deduced.

However, We place related work in presenting explanations alongside news recommendations into the broad categorization of *feature-based*, *similarity-based*, *neighbour-based* and *influence-based* explanations.

### 4.2.1 Highlighting Feature Relevance

Early academic work on feature relevance for generating explanations in news recommender systems have been proposed by Billsus et al.[13] as early as in 1999. Their implementation resulted in explanations highlighting relevant keywords on the form "*This story received a high relevance score, because it contains the words  $f_1$ ,  $f_2$ , and  $f_3$* ", where  $f$  are relevant keywords for a respective news article.

Similarly, Herlocker et al.[39] proposed explanations through the highlighting of feature relevancy in a movie recommender system, and further proposed a similar approach to that of Billsus et al. through combining the most relevant features in the form of textual explanations or sentences.

In light of more recent recommender systems implementing tags for their recommendable items, Vig et al.[45] introduces *tagsplanations*; justification based explanations that utilize third-party generated information source; specifically generated community tags. Vig et al. presented these justifications in a top-n fashion, visualizing the relevance of each tag and comparing them to the preferences in the user profile.

The recent years have seen an increase in research on explanation through feature relevance. For instance, numerous implementations of the explanation technique LIME — first presented by Ribeiro et al.[80] — have been proposed in recommender systems and classifiers alike, through highlighting the top features and their relevancy to the recommendation in question[85, 18].

Similarly, Hove et al.[41] proposed a list-wise explanatory model for explaining rankings in a news recommender system. As we will elaborate upon in the following section, their implementation is solely based on generating descriptions through a post-hoc approach to learning the importance of features.

Furthermore, Fusco et al.[26] proposed an descriptive explanatory model through highlighting the features that contribute the most to the classification output. In addition to simply presenting the most relevant features in a top-n manner, they visualize the feature importance using a sankey diagram as seen in figure 4.1. This presentation technique differs from others in the literature in that they bring the visual aspect to feature relevancy.

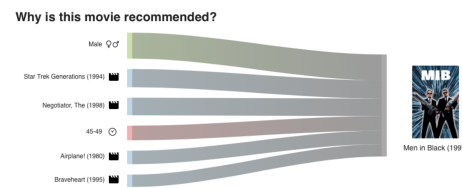


Figure 4.1: A visualization of feature importance using a sankey diagram, highlighting how certain features for a certain user contributed to the recommendation of the movie "Men in Black"[26]

## 4.2.2 Highlighting Similarity

In 2021, Netflix<sup>1</sup> announced their "Play Something" feature, implemented to combat browsing fatigue from navigating through thousands of movies and tv-shows by allowing users to embrace the recommendation engine by letting it choose their entertainment for them. Alongside the newly recommended tv-shows and movies follows a short explanation highlighting the similarity to previously viewed shows alongside some highlighted features. The explanations on the form "This is a [category] similar to [previously watched show]" bear similarity to those of feature relevance, in that the genre is highlighted. However, if the user have not viewed any similar tv-shows or movies, the explanation is reduced to a purely highlighting the feature by stating the genre in a textual manner, i.e. "This is a **family drama** we think you will like".

Recent research covers similarity highlighting to a very small degree, closest one being early works on partial similarity by Symeonidis et al.[91] and the early works by Herlocker et al.[39]. Related work on explanations utilizing user- or item-similarity is mostly implemented in unison with other presentation styles, such as feature relevance or nearest neighbours in explanation models highly coupled with the recommender system. Tintarev et al.[94] mentions that the rationale behind studying user's utilization of item features is that simply stating that two items are similar does not always help users see the commonality between items, while an explanation using feature-based information may better help a user understand how two items are related.

Blanco et al.[14] suggest several explanations in a news recommender system. Their work includes 16 different explanations, in which 5 are justifications that utilize relationships between namely the recommended and recently read news articles, comparing sentences within articles, shared and distinct entities described in the articles and lastly similarities between the geographical locations between articles. Their explanations are purely justifications in that they are not concerned with explaining the underlying recommendation model itself, but rather provide contextual information about the articles and relationships between recommendations and historic interactions.

<sup>1</sup><https://www.netflix.com/>

### 4.2.3 Highlighting Nearest Neighbours

Providing explanations through highlighting nearest neighbours have been implemented in many commercial applications, both in relation to users and items. In general, highlighting neighbours can be divided into simply displaying top-n items, or through displaying items similar to a specific item.

Early work by Herlocker et al.[39] compared the efficiency of different display styles for explanations in user-based collaborative filtering. In this research, the authors provided examples of different explanation styles visualizing how neighbours can be presented as explanations. In the case of users, explanations such as "*This item is recommended to you because a similar user have bought this item before*". In a similar manner, Herlocker et al. demonstrated a textual explanation for items, such as "*This item is recommended to you, because you bought a camera before*". The authors implement both template-based textual explanations as well as visualizations.

In terms of commercial applications, Amazon's "*users who bought x also bought y*" is by many considered the de-facto standard in explaining recommendations through highlighting neighbours in commercial recommender systems.

### 4.2.4 Highlighting Influence

Explanation styles highlighting influence can be viewed as a subcategory of similarity or nearest neighbour styles. Influence style explanations justify the recommendation by isolating the item X that influenced the recommendation Y the most, i.e. "*Item [Y] is recommended because you rated item [X]*".

As with highlighting similarity, pure influence models are rarely implemented by themselves, but rather in combination with others. Early implementations of a movie recommendation system by Symeonidis et al.[91] presented an explanation style combining that of influence and keywords on the form "*Movie [X] is recommended because it contains features [a,b,c...], which are also included in movies [z,w,v...]*".

### 4.2.5 Combining Presentation Styles

Although variations of the methods mentioned are widely implemented in both commercial and academic applications, Bilgic et al.[12] claims that *keyword* and *influence* style explanations are superior to *neighbour* and *similarity* methods, as they allow users to more accurately predict their true opinion of an item. Nonetheless, both *keyword* and *influence* style explanations cannot solely justify the recommendations, as they are solely based on ratings or content.

Fortunately, research on recommender systems implementing CBF and CF in unison have increased since Bilgic et al.[12], allowing the deduction of explanations involving *keyword* and *influence* style explanations in combination[91], presented in the form of a sentence as with sentence explanations mentioned in section 3.4.3.

Such sentence style explanations combining different presentation styles have gained traction during the last couple of years. Li et al.[54] recently proposed a *context-aware*

explanation model based on supervised attention. The model retrieved context as well as contextual features from written user-reviews, producing context-aware feature-level explanations, i.e. "*This product is recommended for you because its [features] are suitable for your current [context]*".

In terms of "where is the limit" for combining presentation styles, a recent study by Kouki et al.[51] research how explanation styles can be combined with hybrid recommender systems in which several information sources are utilized. Additionally, the researchers experimented with manipulating the number of presentation styles such as different visualizations and text. Their research concluded that different styles perform well in combination, however the authors concluded that an explanation should at most include three different presentation styles.

## 4.3 Methods for Explaining Recommender Systems

The following subsections provide an overview of related work on generating, determining or extrapolating information sources to be used in explainable recommender systems.

### 4.3.1 Determining Feature Relevance

Although the approaches for explanations through feature relevance are highly coupled with specific classification models, such explanations can be deduced without classification in mind. For instance, Hovee et al.[41] proposed a list-wise explanatory model for explaining rankings in a news recommender system. Their work was not concerned with providing explanations for a certain classifier, but rather suggesting a way to explain a given ranking of news articles. Firstly, their proposed model learned the importance of individual feature values by changing them and observing how the changes affected the rankings. Furthermore, the model learned the most important features by observing which changes that affected the ranking the most, lastly presenting those features to the user in an understandable way.

Fusco et al.[26] propose RecoNet; a neural recommender system architecture able to retrace the contribution of the original features leading to a given prediction. They use layer-wise relevance propagation (LRP) methodologies to understand the individual contributions of the input attributes, a prominent method in XAI[66]. In short, LRP trace back the contributions from the input layer to the output layer, layer by layer. The activation strength for each class in the final node is conserved per layer. The architecture is a MLP with a softmax activation function on the final node where the target label is an item ID. A user is represented by click history, user attributes and item attributes. Meaning that the users are identified by a set of items and attributes. The items and features are projected in an unified embedding layer.

The increasing interest in classification models based on alterations of ML techniques have suggested the need for *agnostic* explainability models that can be utilized by a variety of ML-based classification models. Many researchers have therefore developed and implemented several model agnostic interpretability tools which quantify or visualize the effects of feature importance[17].

Another interesting contribution to model agnostic explainability approaches is the utilization of a solution concept from cooperative game theory as proposed by Casalicchio et al.[17]. They introduced the Shapley Feature IMPortance (SFIMP) measure, which allows to easily visualize and interpret the contribution of each feature to the model performance. Their proposed methods serve as an evaluation tool that is applied to a data set after a model has been fitted, and allow assessment of feature importance of a fitted model.

Due to its strong axiomatic guarantees, the Shapley values method is emerging as the de-facto approach to feature attribution, with some researchers arguing that it may be the only method compliant with legal regulation such as the General Data Protection Regulation (GDPR)'s "*Right to Explanation*"[1].

There are few implementations of Shapley values in the context of explainable recommender systems in the literature. However, Chen et al.[19] present a framework for layer wise propagation of Shapley values that builds upon DeepLIFT (an existing framework for explaining neural networks) in the context of a medical expert system.

### 4.3.2 Determining Similarity

Billsus et al.[13] conducted early work on providing explanations through an underlying similarity based approach for recommendation explainability. Their system's recommendations are based on scores computed by a hybrid user model consisting of separate short-term and long-term models for representing a variety of interests in different topics. As a consequence, different forms of explanations are used to summarize reasons for a story's relevance. If the story was similar to a previously rated story and therefore classified by the short-term model, the explanation is based on proximity to this previously rated story. The agent retrieves the headline of the closest story in the short-term model that received the same class label as the story whose explanation is to be constructed.

Furthermore, Blanco et al.[14] provide several approaches to determine similarity between entities in a news recommender system.

### 4.3.3 Determining Nearest Neighbours

Determining nearest neighbours for explanations correlates strongly with recommender systems implementing CF as their underlying algorithm. For this reason, approaches for providing explanations through visualizing or highlighting neighbours can be achieved by simply utilizing the neighbourhood in accordance with relevant similarity measures.

Herlocker et al.[39] demonstrates several approaches for item- and user-based explanation through CF-based recommendations. Their conceptual model for user-based neighbourhood explanation was constructed around the baseline approach for user-based CF. The explanations were constructed by simply extrapolating the neighbours by selecting top-n similar users for a given user.

## 4.4 Evaluating Explanations

The literature provides few detailed frameworks on evaluating explanations in recommender systems directly. Tintarev et al.[93] and Zhang et al.[103] both provide some broad categorizations and guidelines for evaluating explanations in recommender systems. In addition, when evaluating explanations in recommender systems one should be aware of potential relationships and restrictions posed on the explanations by the underlying recommender system. As suggested by Tintarev et al.[93] researchers should in some cases consider to evaluate the explainable recommender system as a whole.

As a desirable explainable recommendation model would not only be able to provide high-quality recommendations but also high-quality explanations. As a result, an explainable recommendation model should be evaluated combining both perspectives.

However, due to the scope of this thesis this section will be mainly concerned with evaluating the specific explanations.

### 4.4.1 User Studies

User studies are the most straight forward approaches to evaluating explanations. Such evaluations attempt to evaluate to what degree a certain explanation promotes transparency and trust in a recommendation process. For that reason, subjective perceptions of explanations are often evaluated qualitatively through user surveys with responses typically given on Likert scales[11], statistical scales designed to efficiently capture subjective perceptions of participants.

Early work on evaluating explanations was conducted by Herlocker et al.[39], studying the effectiveness of different explanation styles in a recommender system based on CF. The evaluation involved a user survey, where participants were asked to rank individual movie recommendations in combination with different explanations, and asked to rate the respective recommendation on a scale of one through seven on how likely they were to view the movie. The participants average response on each recommendation were then used for evaluating the effectiveness of the explanation.

While the evaluations conducted by Herlocker et al. considered explainability in respect to its effectiveness alone, Balog et al.[6] provides a framework for eliciting user preferences through surveys grounded in the seven explanation goals identified by Tintarev et al.[93], summarized in table 4.1.

The user surveys are formulated in accordance with the explanation goals, and are based around an item-wise and list-wise evaluation design. Although the item-wise evaluation design has less cognitive load, Balog et al. expect the list-wise design to yield more robust observations, as responses are not influenced by the quality of a single explanation, but rather allows the user to more or less choose the most fitting explanation. Their experiment showed that an item-wise design appears statistically more powerful compared to that of list-wise design, although it would be beneficial to filter for novel recommendations for the recipient. They also found that all seven explanation goals are moderately correlated, with some particular pairs being strongly correlated. Furthermore their experiments revealed that the Satisfaction, Scrutability and Transpar-

ency — if they are desirable goals of a given system — may provide the most complete assessment of explanation quality across the seven goals, further substantiating the assessment by Tintarev et al.[93] when the goals were first formulated.

<b>Evaluation criteria</b>	<b>Definition</b>
Transparency	Explain how the system works
Scrutability	Allow users to tell the system it is wrong
Trust	Increase confidence in the system
Effectiveness	Help users make good decisions
Persuasiveness	Convince users to try or buy
Efficiency	Help users make decisions faster
Satisfaction	Increase the ease of usability or enjoyment

Table 4.1: Evaluation criteria

Furthermore, Vig et al.[45] conducted a user study involving four explanation interfaces on the MovieLens[34] dataset. Subjects evaluated each interface through an online survey, answering questions measuring the role of tag preference and tag relevance in promoting justification, effectiveness and mood compatibility.

Liu et al.[59] conducted a crowd-sourcing evaluation scheme by comparing their proposed model with another state-of-the-art explainable recommender named MLAM[42]. They sampled out the top-100 most active users from the dataset and presented a user's click history and the corresponding items to the worker. Then the worker were asked several questions to compare the recommendations and explanations generated by the model and the MLAM model. The questions were based on Tintarev et al. in table 4.1 and were as follows:

- **Q1:** Which recommendation are you more satisfied with?
- **Q2:** Which model could provide you with more ideas about the recommended item?
- **Q3:** Which recommended item are you more likely to click after receiving an explanation?
- **Q4:** Based on the recommended items, which model generated explanation could help you know more easily and clearly why we recommend it to you?

Q1, Q2, and Q3 are intended to evaluate satisfaction, effectiveness, and persuasiveness, and Q4 were used to evaluate if the attention mechanism is more effective in the proposed method.



### 4.4.2 Online Evaluation

Online experiments can facilitate evaluation of explanations in recommender systems, as they can simulate the natural and familiar environment where users normally encounter recommender systems such as e-commerce, video platforms and online newspapers. Online evaluation scenarios could support several different perspectives, including persuasiveness, effectiveness, efficiency, and satisfaction of the explanations.

However, as pointed out by Zhang et al.[103], measuring the persuasiveness of explanations can prove difficult in online evaluation scenarios, due to the limited type of information that one can collect.

Zhang et al.[104] conducted online experiments aimed at measuring how the explanations affected user acceptance. The authors conducted A/B-tests based on a commercial web browser. Their evaluation was conducted through eliciting three user groups. One receiving the testing explanations, one receiving the baseline "*People also viewed*" and the last one receiving no explanation functioning as a control group. Furthermore the click-through rate of each group was calculated to evaluate the effect of personalized explanations.

### 4.4.3 Offline Evaluation

A variety of offline evaluation methods are suggested for recommender systems. In regards to evaluating explanations, more offline evaluation measures and protocols are yet to be proposed for comprehensive evaluation of explainability.

One approach is to evaluate the percentage of explanations that can be explained by the explanation model, regardless of quality. For this approach, Abdollahi et al.[2] adopted mean explainability precision (MEP) and mean explainability recall (MER), thereby evaluating the top-n recommendations in terms of the explainability of the suggested list.

The authors defined the proportion of explainable items in the top-n recommendation list as *explainability precision* (EP). Furthermore, they defined the proportion of explainable items in the top-n recommendation list relative to the total number of explainable items for a given user as the *explainability recall* (ER). Finally, mean explainability precision (MEP) and mean explainability recall (MER) are EP and ER averaged across all testing users, respectively.

Peake et al.[76] further generalized this through proposing model *fidelity* as a measure of evaluating explainable recommender systems, Model fidelity — as depicted in equation 7.1 — is defined as the percentage of explainable items in the recommended items.

$$\text{Model Fidelity} = \frac{|\text{explainable items} \cap \text{recommended items}|}{|\text{recommended items}|} \quad (4.1)$$

A second approach is to evaluate the quality of the explanations directly. In context of this approach, evaluating the quality of explanations highly depend on the characteristics of the explanations in question[103].

As described in section 3.4.3, some explanation approaches aim to include explanations in the form of textual sentences. Evaluating such sentences can be performed through utilizing state-of-the-art evaluation tools for text generation tools.

Overall, regardless of the explanation style (text or image or others), offline explanation quality evaluation would be easy if we have (small scale) ground-truth explanations. In this way, we can evaluate how well the generated explanations match with the ground-truth, in terms of precision, recall, and their variants[103].



# Data

The following chapter provides an overview of the datasets used for this thesis. Two datasets — one in Norwegian and one in English — will be used throughout this thesis. Section 5.1 discusses available datasets. Section 5.2 and 5.3 provides documentation on each of the two datasets.

Lastly section 5.4 provide a detailed overview of the different stages of pre-processing performed for mitigating shortcomings, further increasing the *recommend-ability* and transfer-ability of the respective datasets.

## 5.1 Available Datasets

Large-scale and high-quality datasets can significantly facilitate the research in an area. There are several public datasets for traditional recommendation tasks such as the Movielens[34] dataset for movies and equivalently the Book-Crossing<sup>1</sup> dataset for books, that have long been considered benchmark-datasets in the recommender paradigm. Due to their large scale and detailed characteristics, many well known recommendation-techniques have been developed utilizing these datasets[100].

News recommender systems have long been restricted to smaller datasets with a variety of quality, scale and characteristics. Recently, Wu et al.[100] and Gulla et al.[31] proposed two large datasets in English and Norwegian respectively to be used for research on recommender systems. Table 5.1 provides a comparison of relevant datasets for news recommender systems.

---

<sup>1</sup><http://www2.informatik.uni-freiburg.de/~chiegler/BX/>

Dataset	Language	# Users	# News	# Clicks	News Information
Plista	German	Unknown	70,353	1,095,323	title, body
Adressa	Norwegian	3,083,438	48,486	27,223,576	title, body, category
Globo	Portuguese	314,000	46,000	3,000,000	embeddings only
Yahoo!	English	Unknown	14,180	34,022	word ids only
MIND	English	1,000,000	161,013	24,155,470	title, abstract, category

Table 5.1: A comparison of available news datasets[100]

## 5.2 The Adressa Dataset

The Adressa Dataset was constructed as part of the RecTech project on recommendation technology by Gulla et al.[31]. The data was extracted using the Cxense platform for news recommendation and monitoring. The dataset covers one week of web traffic from February 2017 on the *www.adressavisen.no* site.

The raw dataset extracted from the Cxense platform is segmented into different folders containing a variety of attributes that are inessential in the context of news recommender systems. Therefore, Gulla et al. constructed a compact version of the dataset tailored for recommender tasks. The compact dataset is roughly comprised of two parts:

1. **Table of reading events**

Each row representing an event, and includes attributes such as event description, article viewed and id representing a unique user.

2. **Table of articles**

Each row representing an article, and includes attributes such as article body, image references and categories.

### 5.2.1 Characteristics

The datasets contains a variety of features, supporting a wider range of recommendation strategies than the public datasets currently in use[31]. Previous datasets largely contain attributes favoring collaborative based recommendation techniques. The adressa dataset compliments this by supporting a variety of attributes especially relevant for research in CBF.

Compared to other datasets such as Movielens[34], the adressa dataset does not deliver explicit user ratings. Instead, it offers a variety of implicit factors that can be used for inferring implicit ratings.

As summarized in table 5.2, the dataset consists of 74,885 news articles, with roughly 27 million interactions from 15,514 users.

# News articles	74,885
# Users	15,514
# Categories	160
# Impressions	27,223,576
Avg. title length	6.61
Avg. abstract length	16.88

Table 5.2: Detailed statistics of the Adressa dataset

## 5.2.2 Articles

The dataset contains a total of 74,885 news articles dating back to 1999. Each article has an *articleID*, a *title*, *description*, *body* and *teaser* in addition range of different keywords and meta-data. All attributes supported by articles are described in greater detail in table 7.3b and 7.3b.

Figure 5.1b and 5.1a show the length distribution of the news title and description. We observe that news-titles are usually very short, with an average length of 6.61 words and a standard deviation of  $\sigma = 3.32$ . Each article also contains a *description* — equivalent to an abstract. The descriptions contains more detailed information about the article, but are also very short, with an average length of 16.8 words and a standard deviation of 8.39.

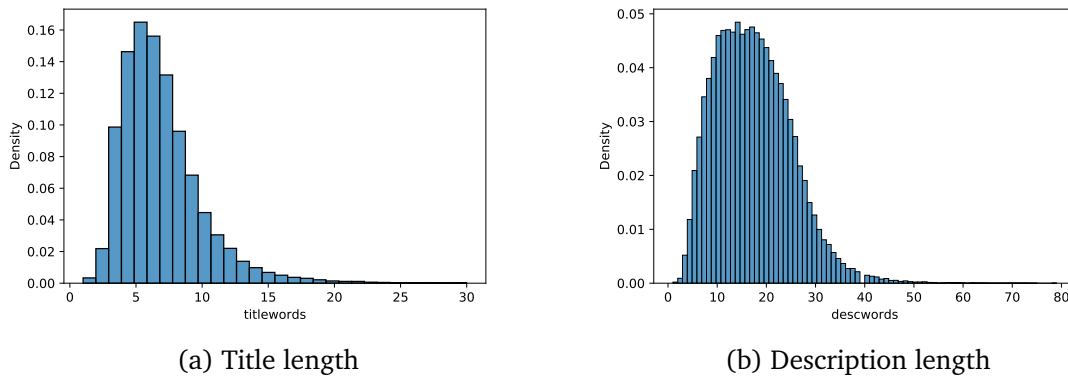


Figure 5.1: Histograms with key statistics of the Adressa dataset.

### 5.2.2.1 Reading-Events

The reading events are comprised of interactions on the adressa.no website. Each row includes 18 attributes describing the event and identifying the user viewing the article. All users interactions are anonymized, but can be identified with the attribute *userID*.

Of the 15,514 users in the dataset, 672 are registered as subscribers, meaning that the user has access to articles hidden behind a pay-wall. A subscriber can also be tracked through individual sessions.

Users known as non-subscribers comprise most of the user-base. In regards to non-subscribers there are two aspects that must be taken in account:

- There might be articles of interest behind the paywall that the user might want to read, but is unable to reach. Hence, the users are not necessarily reading *all* articles of interest to them.
- Every session constructs a new *userID*. Hence, a particular user will be associated with a new ID every time he or she initiates a new session.

Therefore, the user-profile of non-subscribers are considered less complete in the current form of the dataset. Methods from cross-device tracking may alleviate some of these shortcomings.

With a total of 2,717,915 reading-events, the density of the dataset is about 0.19%. Calculating the density per day shows a 0.21% density on most days, although the data for day 1 is very sparse with a density of only 0.11%[31].

Furthermore, the *active-time* attribute records the amount of time spent on the particular article, which in turn can be used for inferring implicit signals of interest to normalize the otherwise binary ratings based based on whether or not a user have clicked an article.

## 5.3 The MIND Dataset

The Microsoft News Dataset (MIND) is an open-source dataset constructed by Wu et al.[100] to facilitate the research in news recommender systems. It was collected from the user behaviour logs of Microsoft News<sup>2</sup>. Its comprised of 1 million users, randomly sampled during 6 weeks from October 12 to November 22, 2019.

### 5.3.1 Characteristics

The MIND-small dataset is a subset of the MIND dataset generated by sampling 50,000 users and their behaviour logs. The dataset is comprised of four parts, *behaviours*, *articles*, *relation-* and *entity-embeddings*. Due to the nature of this thesis, representations such as embeddings will be generated from the ground up, and we therefore restrict ourselves to the behaviours and news articles in the dataset.

#### 5.3.1.1 Behaviours

The authors collected the behaviour logs of sampled users in the collection period, and further formatted these into impression logs. An impression records the news articles displayed when a user visits the web page, in addition to those actually clicked by the user, further constructing labeled samples for each user. The format of each labeled sample is  $[uID, t, ClickHist, ImpLog]$ , where the  $uID$  is the user anonymous ID,  $t$  is the timestamp. The  $ClickHist$  is a list representing the click history of the user, with the article ID of each respective article. Furthermore the  $ImpLog$  contains the articles that have been shown to the user, each with a binary label corresponding to whether or not the user clicked the article.

#### 5.3.1.2 Articles

Each of the 51,281 news articles in the MIND-small dataset contains a news ID, a title, an abstract, a body and a category label — manually tagged by the editors. In addition, Wu et al. extracted rich entities from the title, abstract and body of each respective news article to facilitate the research of *knowledge-aware* recommender systems. Detailed specifics of the dataset can be seen in figure 5.2 and table 5.3

We observe that news-titles are usually very short, with an average length of 10.75 words. In comparison, the abstracts are much longer with an average length of 36.17 words. However, as illustrated in figure 5.2a the length of titles are more or less normally distributed around the average length with a standard deviation of  $\sigma = 3.2$ . Compared to that of the abstract lengths in figure 5.2b resembling a polynomial.

---

<sup>2</sup><https://microsoftnews.msn.com/>



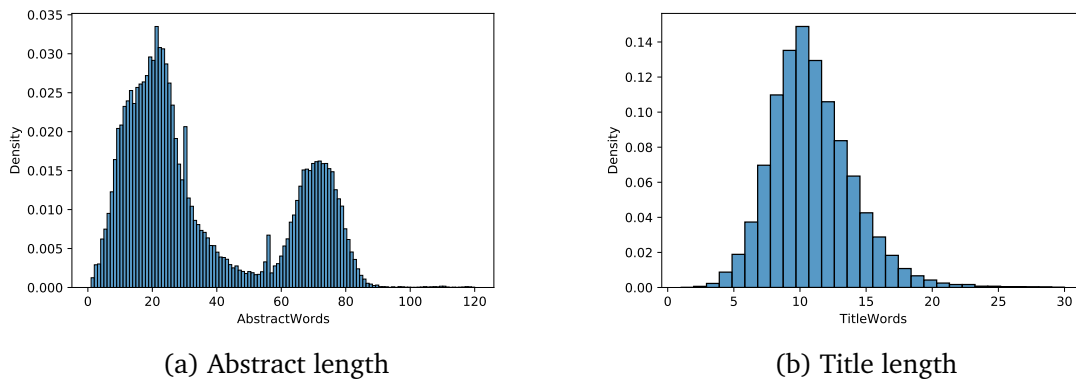


Figure 5.2: Histograms with key statistics of the MIND-small dataset.

# News articles	51,281
# Users	156,964
# Categories	20
# Impressions	156,964
Avg. title length	10.75
Avg. abstract length	36.17

Table 5.3: Detailed statistics of the MIND-small dataset

### 5.3.2 Preprocessing

As underlined by Wu et al.[100], the MIND dataset is already prepared for supporting recommendation tasks out of the box. The authors also supply a variety of out-of-the-box tools that can be used for easily implementing the dataset in a recommendation context.

## 5.4 Attribute Selection

Raw datasets typically provide a large number of assorted attributes. The act of selecting the most relevant attributes for a given data related task is referred to as attribute subset selection. As with both datasets presented in chapter 5, a variety of attributes are supported. Henceforth; the current task of selecting attributes is related to selecting those most relevant for the contributions and models presented in this thesis.

The selection of such attributes can be performed in a structured manner, and although detailed frameworks and approaches for attribute subset selection are proposed in the field of datamining, implementing such a framework is not within scope of this thesis.

The overall goals of the attribute selection performed in this project can be summarized as follows:

- **1.** Ensure that the subsets of both MIND and Adressa are as homogeneous as the common attributes allow for.

- 2. Reduce the complexity in coherence with the research goal and relevant research questions.

Where the first selection goal is concerned with transfer-ability of our proposed models across the two datasets, the latter is concerned with "*holding us back*" and within scope of the project. As both datasets have many shared characteristics and can substantiate a large variety of interesting and motivating research in news recommendation, we are mainly concerned with the task at hand; namely *explaining* recommendations in a news recommender system.

The datasets are therefore prepared in such a fashion that will support a novel neural recommendation engine, and in addition will be able to contribute to generation of explanations alongside the respective recommendations.

### 5.4.1 Items

As introduced in chapter 2, the well renowned issue of a *long-tail* is related to how some recommendable items receive large numbers of ratings or interactions, while the majority receives fewer and in some cases no interactions at all. This issue is further escalated in pure collaborative based recommender systems, as items with few to no ratings are less likely to be recommended.

As both datasets feature large amounts of news articles that have no interactions at all. There can be several reasons to why these articles have no interactions, but through observation this seems to be mainly related to old news articles, and since the sessions were gathered in a short period of time, many news-articles — especially outdated ones — will have few or no interactions. We therefore decided to filter out articles with no interactions, dramatically reducing the size and complexity of the datasets.

This also mitigates the complexity of generating embeddings representative for the news articles, as transformer models such as BERT are known for their notorious time complexity.

We also filtered out news articles that had no header or abstract, such as empty templates and "immediate" news stories such as "*Brann vant seriegull!*" that would later be replaced with elaborated versions. Although such articles can be seen as informative for a short period of time, they provide little leverage for the task of explaining recommendations. More importantly, this also filtered out "dirty" or inconsistent data such as empty news articles and other websites that are naturally unreachable for the users.

### 5.4.2 Users and Interactions

The coverage of interactions between users and items differ in both datasets. Where MIND track individual users and store their browsing history, or previously interacted news articles, Adressa store *all* interactions on their website. This results in a very large number of interactions. These are narrowed down by filtering out interactions that does not correspond to a respective news article, but i.e. corresponds to the front page, navigation menus, categories and such. We can now construct user profiles similar to

those in MIND, where we concatenate all user interactions with news articles that share the same user-id.

### 5.4.3 Notable Observations

As the predefined user-profiles supplied by MIND provide 'out of the box' data ready for implementation in a recommender system, it does not contain any specific implicit tracks laid by the user. In contrast, many interactions supplied by Adressa include time spent on the webpage. We considered utilizing this information for improving the efficiency of the recommendations through normalizing the ratings by word-count divided by read-time compared with average words read by an adult as suggested by Mitchell et al.[64]. However, this was discarded as it is not applicable with the MIND dataset in addition to being out of scope.



# Method

There are many components involved in developing an explainable recommender system. This chapter describes the proposed ENSUS model for providing explanations in a news recommender system, and describes each component in great detail. The proposed model is based on research on the current state-of-the-art in explainable recommender systems with respect to the taxonomy described in chapter 3.

Firstly, section 6.1 provides a conceptualization of ENSUS and provides an architectural overview of the task at hand. Secondly, section 6.2 describes the individual components in detail. Based on the descriptive nature of ENSUS, we provide a second approach to justification described in section 6.3.

## 6.1 Conceptualizing the ENSUS Model

Commercial state-of-the-art recommender systems that provide explanations often provide little transparency to the recommendation process. For example, commercial explanations as those implemented by Amazon, Netflix and Instagram are very general in the case of "*users who purchased this item also purchased...*" or "*since you watched item ... we believe you like ...*". In session-based e-commerce applications, such explanations are reliant on the user having an understanding of the relationship between the viewed items and its recommendations, thus adding to the cognitive load. Topic based approaches (section 3.5.2) attempts to solve this issue by providing a summarized contextual version of the user's click history and then compare the summary with the topic of the recommended item. For example: "*since you like romantic movies, we recommend Titanic*".

As conceptualized by Vig et al.[45], a recommendation explanation may be one of two types: a *description* or a *justification*. Descriptions reveal the actual underlying mechanisms that generates the recommendations, while justifications convey a conceptual model that may very well differ from the underlying mechanisms, revealing other relevant aspects of the recommendation process such as common characteristics between the recommended article and those previously read.

For our proposed method, we make an attempt to combine the two dimensions by providing a combination of topic based explanations with the feature importance of the topic that capture the user's preferences. Here, the topic based explanation serves as a *justification* while the feature importance serves as the *description*. Specifically, the proposed method provide explanations by highlighting feature relevancy through local Shapley values. For the purpose of this thesis, we name the proposed method ENSUS which is an acronym for *Explainable Neural recommender System Using Shapley values and topic modeling*.

ENSUS is designed after the explanatory criteria in recommender systems by Tintarev, described in section 4.4.1. Furthermore, it is designed to reduce the cognitive load by using topic modelling as mentioned above. Specifically, the model should be able to:

- Increase transparency by helping the user to understand how the recommendations were predicted by forcing the model to recommend items that match the user's preferences.
- Increase transparency with the use of feature relevance though Shapley values.
- Provide scrutability by allowing the user to influence future recommendations by altering its own user profile.
- Increase trust by leveraging accurate recommendations and show that user preferences are present in the recommendations.
- Be satisfying to use as it has low cognitive load, provide transparency, scrutability, and trust.

To further conceptualize ENSUS we present the following example, illustrated in figure 6.1. Consider a user U1 with the inferred preferences *Celebrities*, *Movies* and *Golf*.

In step 1 in the figure, the user is presented with one recommendation which matches two topics in her user profile along with an explanation which tells the user how much each of the two topics contributed to the recommendation.

In step 2, the user has removed the inferred topics *Celebrities* and *Movies* and is now recommended an article about golf since the topic golf is the topic present in the user profile.

Note that the user profile and collaborative filtering can be seen as opposites. Collaborative filtering considers the user's history and attempts to model the user's preferences into a predefined number of latent factors. As opposed to collaborative filtering, the user profile can be used to reflect a user's preferences at a given moment without considering his click history. Continuing with the aforementioned example, the click history implies that the user likes golf. If the user now removes golf from his user profile and adds foreign politics, the click history and the user profile is now opposites. As a result, for a period of time, the user profile and the hidden latent factors in CF may be completely opposites. The latent factors may model the user's preferences towards golf while the user profile models the user's preferences towards foreign politics.

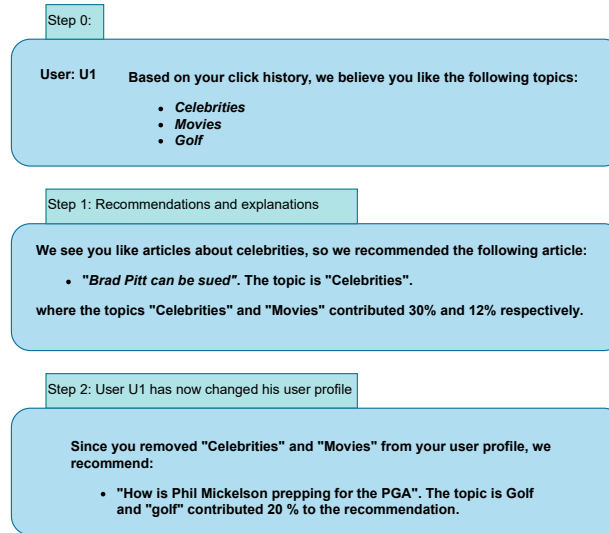


Figure 6.1: Conceptualization of ENSUS.

## 6.2 Proposed ENSUS Model

In the following we present the three core components of the recommendation architecture illustrated in figure 6.2.

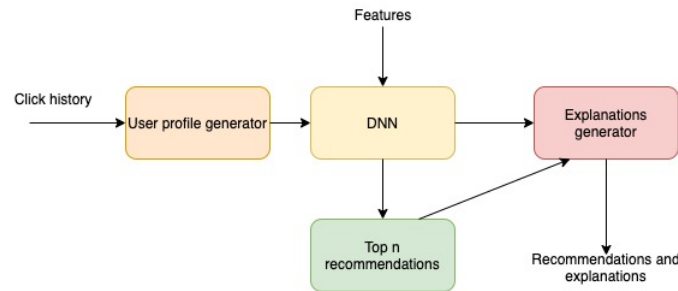


Figure 6.2: Overall architecture of ENSUS.

### 6.2.1 User Profile Generator

The goal of the *user profile generator* is to capture a user's preferences into a set of predefined news topics. Given the set  $G$  consisting of all possible news topics in the news article database, a *user profile*  $P_u$  for a user  $u$  is a subset of  $G$ . The output of the component is an array consisting of  $n$  topics. For example, consider a user  $U1$  who have read news articles about baseball, golf, foreign politics and national politics. Then the user profile for user  $U1$  is:

$$P_1 = ["baseball", "golf", "foreignpolitics", "nationalpolitics"]$$

To capture a user's preferences it simply take the  $k$  most frequently appearing topics in the user's click history and set the top  $n$  elements in the sorted array as the user profile. The user profile generator can be further improved by using more sophisticated methods such as tf-idf, LDA or topic modeling like those described in section 3.5.

## 6.2.2 Recommender Component

Figure 6.3 presents the neural network architecture. The network consists of a *user encoder*, the *news encoder* and the *click predictor*, which are inspired by NPA [99], NeuMF [37] and Wide & Deep [22]. The *user encoder* aims to learn user representations based on the user’s click history. The *news encoder* aims to learn the article representations and the *click predictor* aims to predict the how likely a user will click on a specific article in the future. A thorough justification for the neural architecture is presented section 7.5.

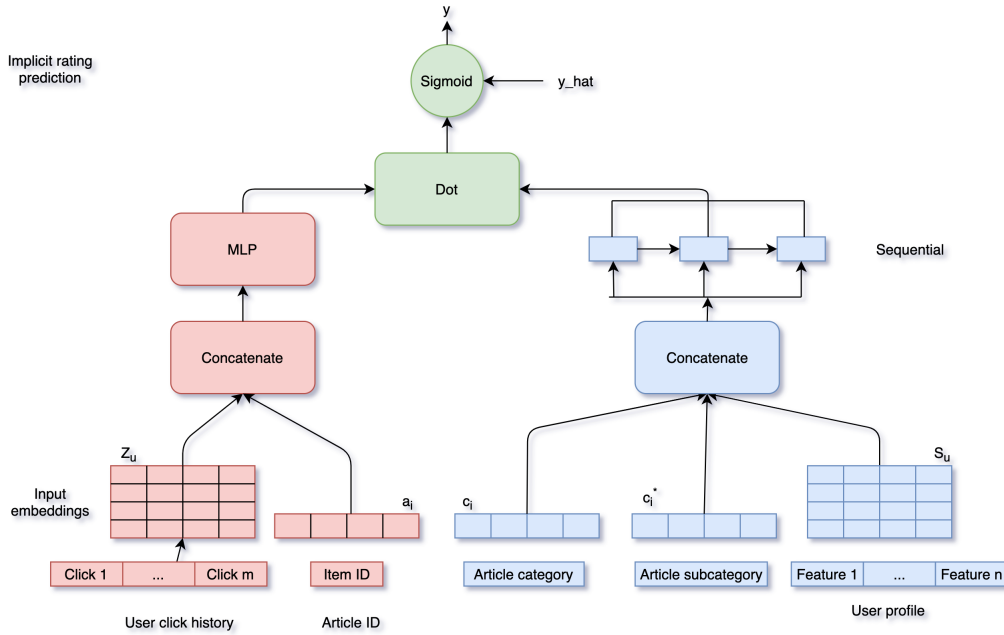


Figure 6.3: Architecture of the neural network for the proposed method

### User Encoder

Given a user’s click history and an article ID, the user encoder aims to learn what articles fits into a user’s click history.

A user  $u$ ’s click history is transformed into the embedding matrix  $Z_u = \{z_{u0}, z_{u1}, z_{u2}, \dots, z_{uf}, \}$ , where  $z_{uf}$  denotes the embedding vector for user  $u$ ’s  $j$ -th article click. An article is transformed into the vector representation  $a_i \in \mathbf{R}^d$  for item  $i$ , where  $d$  is the dimension of the embedding vector. An entry in a embedding matrix is a mapping of a discrete variable to a vector of continuous numbers. The two embedding matrices can be seen as the user and item latent factors respectively. Values in the embeddings are changed and learned during the training phase. Embedding matrices are alternatives to one-hot encoding. One-hot encoding have the drawbacks of high dimensionality for high-cardinality variables. Using one-hot encoding also results in loss in information as similar items are not placed closer to each other in the embedding space.



$Z_u$  is flattened and concatenated with  $a_i$  resulting in the vector  $g_u$ . To learn the user combinations, the vector is sent through fully connected nonlinear layers:

$$q_{uf} = \text{Relu}(w_v \times g_u + b_v) \quad (6.1)$$

where  $w_v$  is the weight vector and  $b_v$  is the bias.

### Item Encoder

The item encoder takes the article category, article subcategory and the user profile and converts the input into two vectors and one matrix;  $c_i \in \mathbf{R}^d$ ,  $c_i^* \in \mathbf{R}^d$  and  $S_u = \{s_{u1}, s_{u2}, \dots, s_{uk}\}$  respectively.  $s_{uh} \in \mathbf{R}^d$  is the h-th topic describing user u. The user profile embedding is flattened and concatenated with category and subcategory vectors into the vector  $s'_{ui}$ . Since the user profile is an important component of the method, we employ a Long-Short Term Memory (LSTM) over the vector representation to include the information for its predecessors over each category and subcategory. Based on the equations in 2.4.6.2 the item encoder returns the vector  $p_{ui}$ .

### Click Predictor

The click predictor takes the user representation vector and the item representation vector as input. In order to combine the representations into a prediction, we use the dot product. The probability of a user clicking an item i is obtained by:

$$\hat{y} = \text{sigmoid}(q_{uf}^T \cdot p_{ui}) \quad (6.2)$$

## 6.2.3 Explanations Generator

As presented in the previous section, ENSUS builds a user profile up front capturing a user's preference, and then generate predictions based on the user profile and other features. Shapley values are used to represent the importance of each preference in the user profile towards the recommendations, and can thus be categorized as a *information source*. Furthermore, these feature values can be presented to the user to describe the recommendation process.

The neural architecture is specifically designed to push the importance of the user profile in order to make sure that a user's preferences is present in the top k recommendations.

### 6.2.3.1 Propagating SHAP Values

The goal of SHAP is to explain the prediction for any instance  $x_i$  as a sum of contributions from its individual feature values.

The explanations generator takes as input the trained model, a users profile and article content and outputs how much each element in the users profile contributed to the prediction. The explanations generator can also be extended to take what ever desired input features and output their contribution to the prediction.

The explanation generator estimates the contributions of each feature value to the pre-

diction. The Shapley value of a feature value is defined as:

$$\phi_j(val) = \sum_{S \subseteq \{x_1, \dots, x_p\} \setminus \{x_j\}} \frac{|S|!(p-|S|-1)!}{p!} (val(S \cup \{x_j\}) - val(S)) \quad (6.3)$$

where  $S$  is a subset of the possible features the model can take,  $val(S)$  is the prediction for the feature values in  $S$ ,  $x$  is the vector of feature values and  $p$  is the number of features [65].

The experiments in this thesis uses KernelSHAP, in which LIME and Shapley values are combined. KernelSHAP estimates the Shapley values using a weighted linear model. Instead of permuting over all possible values in  $x$ , KernelSHAP performs twenty permutations. The output of KernelSHAP is an array of Shapley values where the  $n$ th element corresponds to the  $i$ th input feature in the model. In other words, the input features to the model does not need to match the input to KernelSHAP.

Figure 6.4 illustrates the output of KernelSHAP. *Base value* (0.6194) is the mean value of the model over the entire input space.  $f(x)$  is the prediction value for the input instance. The bars in red illustrates the three features in which contributed towards a higher prediction value and the blue bars illustrates which features contributed towards a lower prediction value.

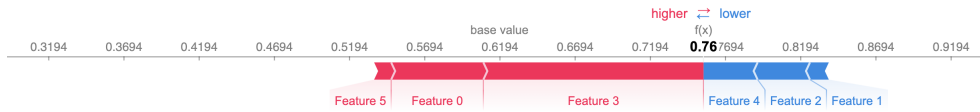


Figure 6.4: SHAPforce\_plot [61]

## 6.2.4 Learning Patterns

Now that we have introduced the architectural design the following presents a discussion on training the recommender component.

Loss optimization is the process of minimizing the value of the loss function. A loss function is used to tell the network whether the prediction is right or wrong. Since we chose to approach the recommendation problem as a binary classification task, meaning the given article is either relevant or irrelevant to a given user, a natural choice is to squeeze the predicted value between 0 and 1 following a probabilistic distribution. We use binary cross entropy to evaluate the error of the prediction:

$$L = -y \cdot \log(\hat{y}) + (1 - y) \cdot \log(1 - \hat{y}) \quad (6.4)$$

where  $\hat{y}$  is the prediction value in the output and  $y$  is the target value. Binary cross entropy compares each of the predicted probabilities to target labels which is either 0 or 1. It penalizes the instances based on the distance from the target value. The loss increases as the predicted value diverges from the target value.

We use Adam [47] for training. Adam is an optimization algorithm used to update the weights of the network during training. It is an alternative to stochastic gradient

descent, described in section 2.4.3. The authors of Adam describe the advantages of Adam in two steps:

- **Adaptive Gradient Algorithm** (AdaGrad): the learning rate adapts to the parameters. It performs smaller updates for parameters associated with frequently appearing features and larger updates for infrequent features. For this reason, it is well-suited for sparse data<sup>1</sup>.
- **Root Mean Square Propagation** (RMSProp): the learning rate decays as the gradients decays.

To sum up, the learning rate will start out relatively high and decay as the algorithm converges. As a result the training time will reduce and save memory.

---

<sup>1</sup><https://ruder.io/optimizing-gradient-descent/>

## 6.3 Entity Similarity

In addition to the proposed ENSUS built on SHAP described in the previous section, we will implement a similarity based model for justifying the recommendations. While descriptive explanations such as feature relevancy through SHAP provides transparency to the recommendation model, justifications allow freedom and flexibility in generating explanations.

We propose a novel approach to explaining news recommendation through entity similarity, where the recommendation is justified based on highlighting the similarity between the recommendation and recently visited articles. The approach is reliant on our hypothesis that the neural recommender will recommend items that are somewhat similar to what the user previously viewed.

Similar approaches have been previously implemented in recommender systems implementing CBF, as they often utilize nearest neighbours of items to recommend similar items. However, in the context of neural classification or CF there is no guarantee that the recommended item may bear strong similarity to the historic interactions.

### 6.3.1 Proposed Framework Overview

Figure 6.5 presents an architectural overview of the proposed model. As depicted, ENSUS is used for generating the recommendations.

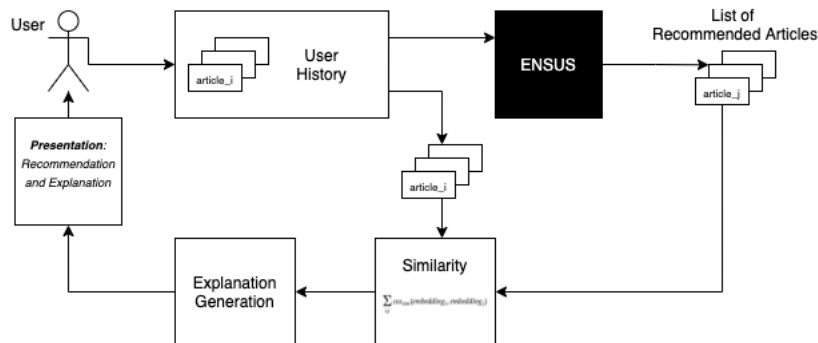


Figure 6.5: A high level architectural overview of the proposed justification by entity similarity, or relationship between read and recommended articles. As depicted

As depicted, the recommendation generation with ENSUS is fully omitted and considered a black box. The model takes only the resulting list of top-n recommendations into consideration. In the "similarity" stage, the embeddings from the recommendations and user profile is compared. If the similarity between a recommendation and that of an historic article is above the cosine similarity threshold of 0.6, a textual explanation is presented alongside that recommendation to the user in a textual fashion.

### 6.3.2 Generating Embeddings

Evaluating the semantic relationship between articles was performed through generating sentence level embeddings from all abstracts. The recently developed sBERT<sup>2</sup> framework by Reimers et al.[79] was utilized, as it is highly efficient and reliable for sentence level comparison tasks. The framework is published as a huggingface package called, *sentence\_transformers*<sup>3</sup>, and is based on the huggingface *Transformers*<sup>4</sup> and *PyTorch*<sup>5</sup> libraries.

Due to the availability of a pre-trained BERT based transformer model tailored semantic comparison of English sentences, we will exclusively conduct experiments related to abstract entity similarity on the MIND dataset.

### 6.3.3 Information Source and Presentation Style

Based on the taxonomy proposed in chapter 3, the information source for the similarities are that of *knowledge about similar alternatives*, in that it compares the recommendation to previous interactions. In addition the approach will utilize *background data*, that is the category of the recommended item, inspired by the explanations introduced by Netflix, described in section 4.3.2.

In regards to presentation style, the explanation will be limited to *template-based textual explanations*, due to the demonstrated efficiency of such explanation styles as compared to visualization[51].

### 6.3.4 Inferring Similarity

Recent work by Lin et al.[57] involves the generation of explainable tuples, or opinion aspect pairs generated based on user provided reviews. The authors determine the semantic relatedness between different tuples for inferring their explanations. In a similar manner, we utilize the sentence embeddings for each article abstract for comparing the semantic relatedness between news articles, thus providing an information source of *relationship between knowledge objects*, as categorized in the taxonomy.

However, since the embeddings are learned from abstract texts instead of e.g. article categories, we have to determine a threshold of similarity where the accuracy of related articles is not compromised, while at the same time ensuring that the articles used in the justification actually share some characteristics.

Lin et al. set a cosine similarity threshold at 0.8, based on a number of observations of tuples with different similarity scores. Similarly, we generated similarity scores for each recommendation in relation to all articles present in the current user history, and observed that most pairs with a cosine similarity above 0.6 had common categories, and broadly discussed related topics. Our observations were persistent through development in which a number of unique, randomly sampled user profiles were observed.

---

<sup>2</sup><https://www.sbert.net/>

<sup>3</sup><https://huggingface.co/sentence-transformers>

<sup>4</sup><https://huggingface.co/transformers/>

<sup>5</sup><https://pytorch.org/>

For this reason, we found that setting the cosine similarity threshold at 0.6 for the MIND dataset can generally distinguish related articles. In addition, we theorize that the overhead has a positive effect on increasing the model fidelity, as we have no guarantees that the ENSUS will recommend articles that are closely related to those previously read.

For this reason, the legitimacy of the proposed model should be demonstrated with respect to the model fidelity[76], or percentage of explainable items amongst the recommendations.



# Experiments and Results

This chapter covers the experiments and results related to the underlying recommendation mechanism and the corresponding explanations.

First, section 7.1 presents an overview of the experimental plan and methodology for evaluating the proposed models. Experimental settings are presented in section 7.2 followed by the quantitative experiments in section 7.3. The qualitative experiments are reported in 7.4. Section 7.5 present the experiments about how the user profile is used to support scrutability and trust. Section 7.6 presents the performance experiments. Lastly, section 7.7 consider some notable observations we made while conducting and evaluating the experiments.

## 7.1 Experimental Plan

In order to answer and substantiate reasoning for the RQs presented in section 1.3 the experimental plan was developed and executed. The experimental plan can be classified with respect to four separate stages:

1. **Quantitative Experiments** - The quantitative phase is concerned with evaluating and comparing the two proposed methods against baselines in a quantitative manner using a user survey.
2. **Qualitative Experiments** - The qualitative phase is concerned with examining the qualitative aspects of the explanatory models and the information sources.
3. **Scrutability Experiments** - The scrutability experiments is an attempt to quantitatively assess whether a user can manipulate her user profile to fit future recommendation to a change in her preferences.
4. **Performance Experiments** - The performance experiments are concerned with evaluating the underlying recommender system of ENSUS. The performance is not the priority of this thesis, however, as mentioned in section 3.6.2.3, trust is highly correlated with the accuracy of the recommendations.



## 7.2 Experimental Settings

We define a set of  $n$  users  $U = \{u_1, \dots, u_n\}$  and a set of  $m$  items  $V = \{v_1, \dots, v_m\}$ . For a user  $u$  we have a preference  $r_{uv}$  which is inferred from an interaction on item  $v$ . As a result, the  $n \times m$  rating matrix  $\mathbf{R}$  is binary.

Following previous work [37, 100], we divide the data into training, validation and test sets using a time-based *leave-one-out* approach. For each user, the last item interaction is held out and put into the test set. Since it is too time consuming to rank all items for every user during evaluation, we randomly sample 99 items that the user has not interacted with and put these items into the test set. For a dataset consisting of  $N$  users, the test set is comprised of  $N$  positive articles and  $99 \cdot N$  negative articles. The remaining data is put into the train set except for 10% which is held out for validation.

To provide negative instances we sample 4 articles for each interaction. The negative samples are fetched from the set of articles the user had not interacted. Finally, we remove users which have less than 5 interactions.

To ensure a robust and sound evaluation environment, we adopt the two datasets introduced in chapter 5 for all experiments if nothing else is specified.

Code for the experiments are available as open source on GitHub<sup>1</sup>

### 7.2.1 Parameters and Hyperparameters

In order to evaluate each recommender on a fair ground, every neural recommender have the same set of hyperparameters if nothing else is specified. Due to problems with overfitting on all baselines, we adopt the regularization techniques: embedding initialization, dropout, batch normalization and l2-regularization.

<b>Learninig Rate</b>	0.01
<b>Embedding Initializa- tion</b>	he-normal
<b>Emebedding Dimen- sions</b>	20
<b>Dropout</b>	Ranging from 0.2 to 0.8 depending on the number of neurons in the layer.
<b>Batch Normalization</b>	Used after LSTM layers and deep MLPs
<b>Optimization Al- gorithm</b>	Adam
<b>Loss Function</b>	Binary Cross Entropy
<b>l2-regularization</b>	Used on all embeddings and deep MLPs

Table 7.1: Parameters and hyperparameters

<sup>1</sup><https://github.com/EivindFa/ENSUS>

## 7.3 Quantitative Evaluation

To quantitatively evaluate our models explainability, we conduct crowd-sourcing evaluation by comparing our proposed models to five baseline methods further elaborated upon in section 7.3.2.

We adopt the item-wise experimental evaluation framework proposed by Balog et al. [6] from Google for quantitatively evaluating each explanation. The framework incorporates the seven goals of explanations originally proposed by Tintarev et al. [93], depicted in table 7.2. The recommendations depicted in the user survey were generated using ENSUS recommendation component on the MIND dataset.

<b>Evaluation criteria</b>	<b>Definition</b>
Effectiveness	Help users make good decisions
Efficiency	Help users make decisions faster
Persuasiveness	Convince users to try or buy
Satisfaction	Increase the ease of usability or enjoyment
Scrutability	Allow users to tell the system it is wrong
Transparency	Explain how the system works
Trust	Increase confidence in the system

Table 7.2: Evaluation criteria or explanation goals as proposed by Tintarev et al.[93]

### 7.3.1 Survey Overview

The subjects are asked to rank each explanation with respect to the seven explanation goals on a five point Likert scale[11], where subjects rank each explanation based on their level of agreement to a statement representative of each explanation goal. The wording of each statement was chosen with respect to the wording and findings presented in Balog et al.[6], and is depicted in table 7.3.

Furthermore, the survey consisted of four parts:

1. A short description of the overall goal of the survey, how the survey will be conducted as well as privacy related information.
2. A presentation of the historic interactions / previously read articles of the sampled user profile.
3. An item recommendation (item-wise design) accompanied by an explanation.
4. Seven statements presented in a random order each targeting a respective explanation goal.

For each subject, part (3) and (4) were repeated eight times, each showing a different explanation for the same recommended item. This naturally resulted in a large amount of questions, and we therefore determined to exclude reverse statements to avoid survey fatigue.

<b>Explanation Goal</b>	<b>Evaluation Statement</b>
Effectiveness	This explanation helps me determine how well I would like the article.
Efficiency	This explanation makes me more effective when reading news articles.
Persuasiveness	This explanation makes me want to read the article.
Satisfaction	This explanation would make it easier to pick recommended articles.
Scrutability	This explanation would allow me to provide concrete feedback on whether or not my preferences have been understood.
Transparency	This explanation helps me understand what the recommendation is based on.
Trust	This explanation increases my trust in the recommendation.

Table 7.3: The seven evaluation statements with their corresponding evaluation goal.

The user subjects were 30 Norwegian participants. All subjects were active news readers, with an age span between 22 and 35. Due to applicable GDPR regulation, no additional information will be revealed about the participants. The user survey can be viewed in its entirety in Appendix 8.2.5 and the results will be presented in section 7.3.3.

### 7.3.2 Baselines

In addition to the proposed methods, we include five baseline explanations shown in table 7.4. Amongst the baseline models is a "lazy" simplistic approach of simply justifying the recommendation with "*We think you would like this article*".

In addition, its worth noting that although the implementation proposed by Wang et al.[97] utilizes a knowledge graph for inferring the entity relationship, implementing such a model is beyond the scope of this thesis. However, the explanations demonstrated by the authors can be performed by utilizing the abstract entities present in the MIND dataset. However, the qualitative aspects of the explanatory model is restricted to the quality of the pre-generated entities.

Baseline Explanations	Description	Explanation
<i>TargetSnippet</i> [14]	This explanation is constructed by simply taking the first two sentences appearing in the recommended news. In our case we choose to include the whole abstract, as it contributes with more context and has a comparable length to that of two sentences.	"As the impeachment inquiry intensifies, some associates of the president predict that his already erratic behavior is going to get worse..."
<i>SharedEntity</i> [14, 97]	The explanation tells the user that a given named entity $X$ is shared between read and recommended news.	"This article is about newsopinion, which is amongst your interests."
<i>RippleNet</i> [97]	Use a knowledge graph as side information to extend user preferences. Explanations can be generated by traversing the edges in the graph.	"Because you read "News in Cartoons", which also mention Donald Trump."
<i>HighlightingCategory</i>	The explanation tells the topic of the recommended article.	"This is a newsopinion article."
<i>Substantiation</i>	The explanation only involves labeling the explanation as a recommendation	"We think you would like this article."

Table 7.4: Baseline explanations inspired by related work

Name	Type	Information Source	Explanation Model	Presentation Style
ENSUS Visual	Description	Decisive input values	SHAP	Chart-based visual
ENSUS Sentence	Description	Decisive input values	SHAP	Template-based textual
Similarity	Justification	Relationship between knowledge objects, Background data	Abstract embedding relationship	Template-based textual
Ripplet	Justification	Relationship between knowledge objects	Abstract entity relationship[97]	Template-based textual
Abstract Snippet	Justification	Background data	Target Snippet[14]	Template-based textual
Shared Entity	Justification	Relationship between knowledge objects	Shared Entity[14]	Template-based textual
Highlight category	Justification	Background data	N/A	Template-based textual
Substantiation	Justification	N/A	N/A	Textual

Table 7.5: An overview of all explanations with respect to their type, information source, explanation model and presentation style.

### 7.3.3 Results

The following subsections provide an overview of the quantitative results for each respective explanation. Each subsection includes two visualizations of the responses, namely a stacked Likert visualization for visualizing each response, and a correlation matrix. Due to the ordinal nature of the Likert data, correlations are calculated using Spearman correlation.

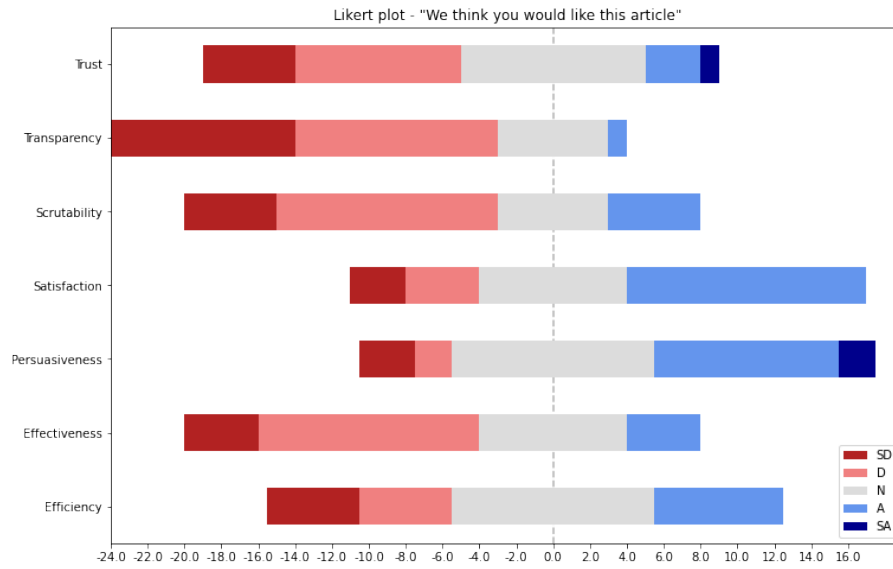
For the stacked charts, each color is representative of a Likert scale point with corresponding colors depicted by the legends. In the case of the legends, the abbreviations are as follows: *SD* stands for *Strongly Disagree*, *D* stands for *Disagree*, *N* stands for *Not Sure*, *A* stands for *Agree* and lastly *SA* stands for *Strongly Agree*.

#### 7.3.3.1 Substantiation: Simple Textual Justification

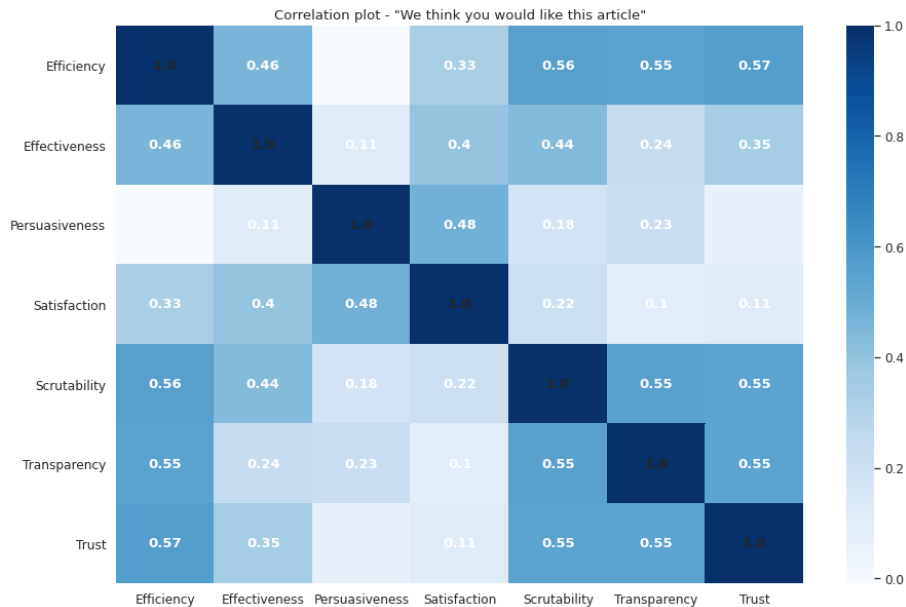
We had low expectations to this explanation, as it essentially only involves labeling the recommendations as recommendations, but with a personal appeal; hence the wording of "We think you would like this article".

Figure 7.1 depicts visualization of the results in relation to each explanation goal, hereby a stacked visualization of the Likert scores, in addition to a correlation matrix.

Right away we observe that the explanation scores very low in terms of transparency and trust, suggesting that the explanation provide little to no transparency to the recommendation process. Interestingly, the explanation scored somewhat well on satisfaction and persuasiveness. This might indicate that the candidates are familiar with the accuracy of recommender systems, and that potentially the subject was persuaded by the fact that the recommendation was labeled as a personalized recommendation.



(a) Stacked visualisation of the Likert scores



(b) Correlations related to each goal

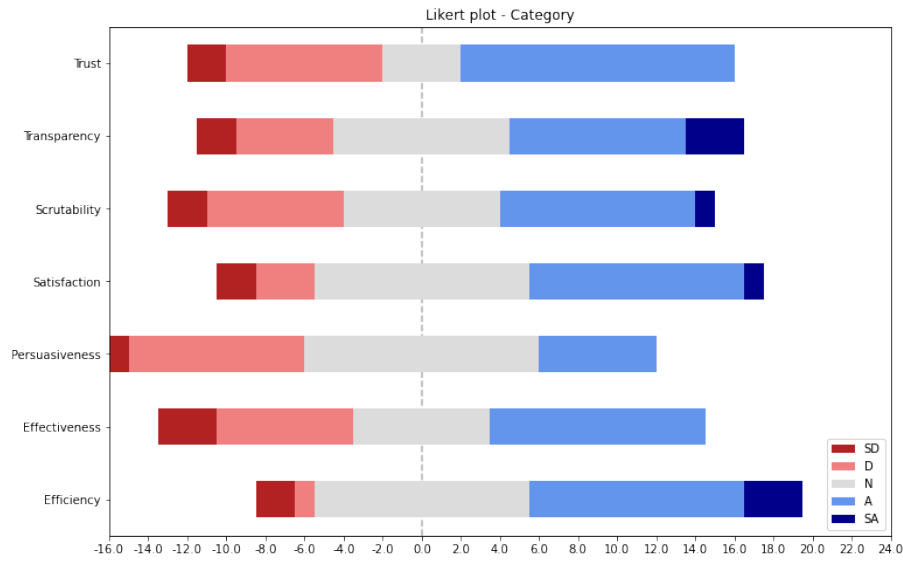
Figure 7.1: Results on explanation through recommendation substantiation.

### 7.3.3.2 Highlighting Category

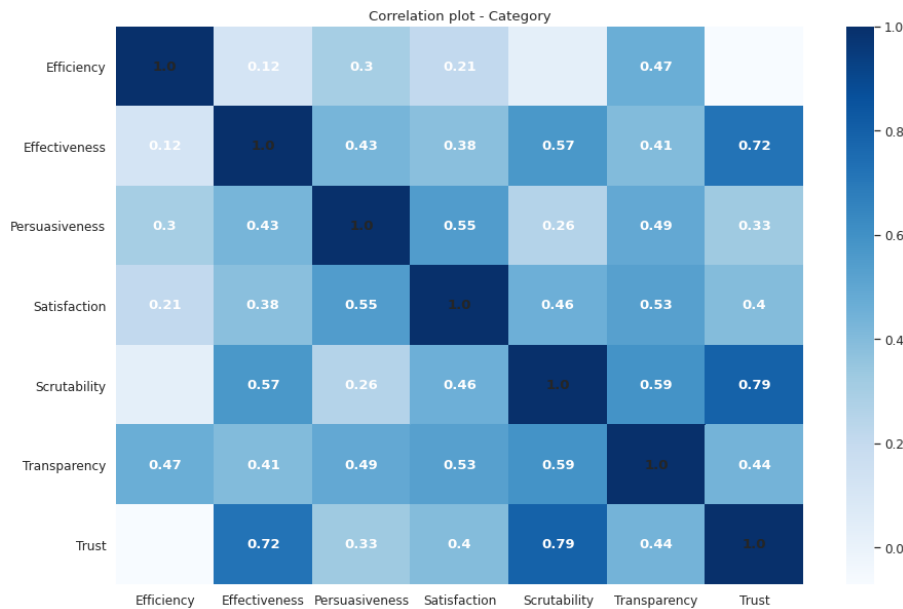
The second explanation is a description involving only highlighting the category of the news article, thus contributing with additional information about the recommendation other than its title.

Figure 7.2 depicts visualization of the results in relation to each explanation goal, hereby a stacked visualization of the Likert scores, in addition to a correlation matrix.





(a) Stacked visualisation of the Likert scores



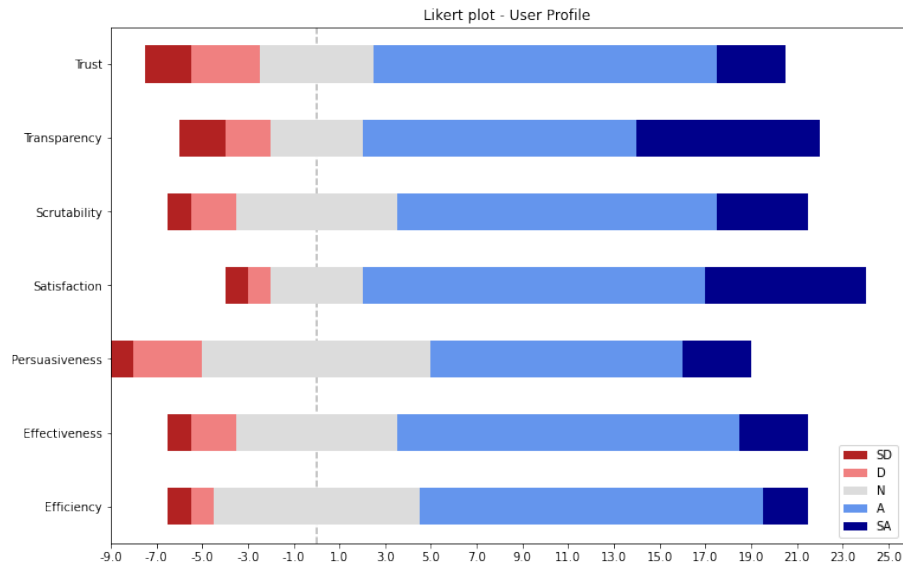
(b) Correlations related to each goal

Figure 7.2: Results on explanation through highlighting the news category.

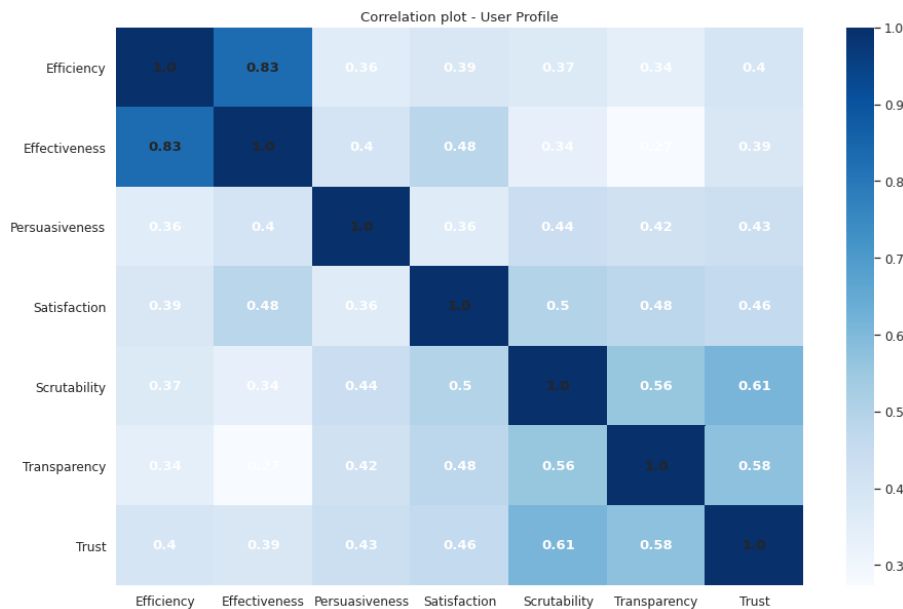
### 7.3.3.3 Shared Entity

Figure 7.3 depicts visualization of the results in relation to each explanation goal, hereby a stacked visualization of the Likert scores, in addition to a correlation matrix.

We observe notably high scores on satisfaction, and we note a high correlation between efficiency and effectiveness, implying that the subjects that find the explanation helpful in determining their potential interest in the recommendation also become more efficient when reading news articles.



(a) Stacked visualisation of the Likert scores

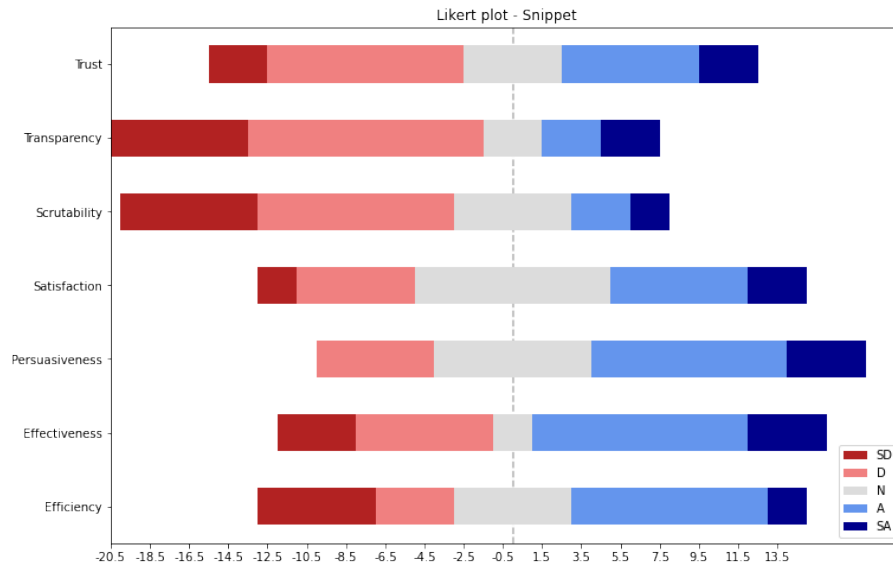


(b) Correlations related to each goal

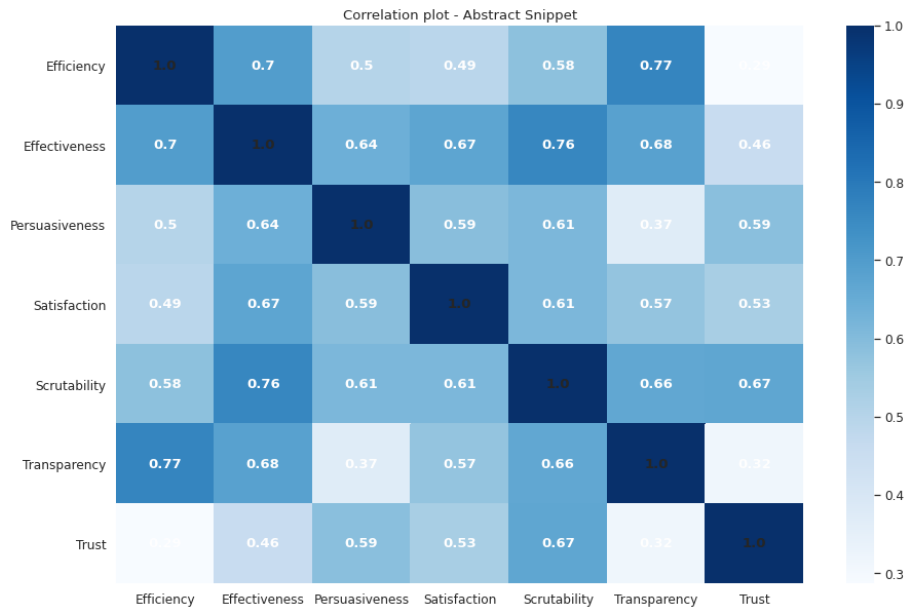
Figure 7.3: Results on explanation through conformity of news category and user profile, or *shared entity*[14]

### 7.3.3.4 Abstract Snippet

Figure 7.4 depicts visualization of the results in relation to each explanation goal, hereby a stacked visualization of the Likert scores, in addition to a correlation matrix.



(a) Stacked visualisation of the Likert scores

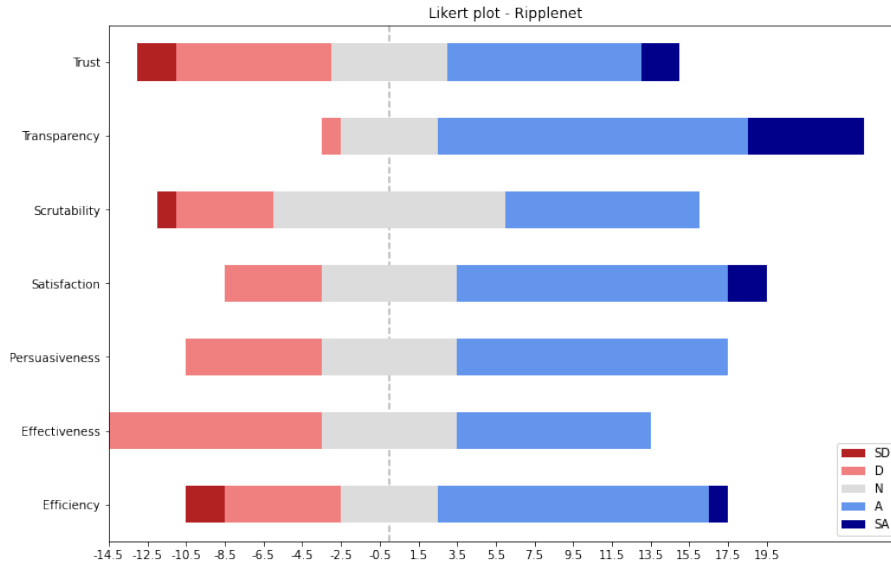


(b) Correlations related to each goal

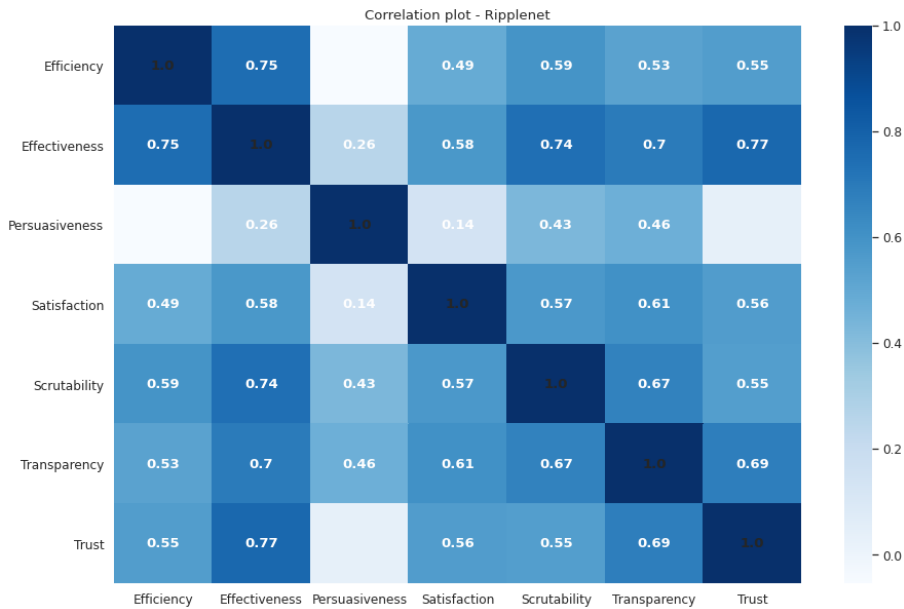
Figure 7.4: Results on explanation through abstract snippet[14]

### 7.3.3.5 Ripplenet

Figure 7.4 depicts visualization of the results in relation to each explanation goal, hereby a stacked visualization of the Likert scores, in addition to a correlation matrix.



(a) Stacked visualisation of the Likert scores

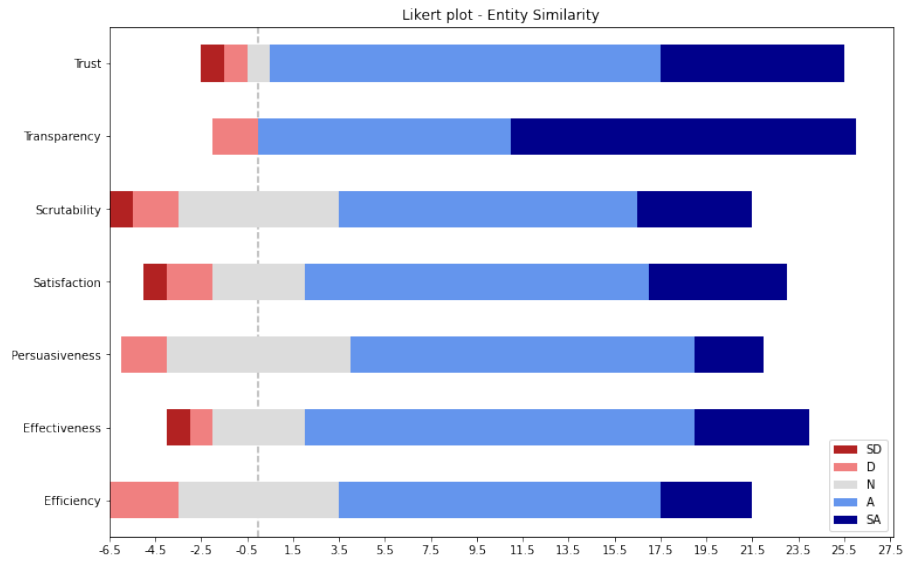


(b) Correlations related to each goal

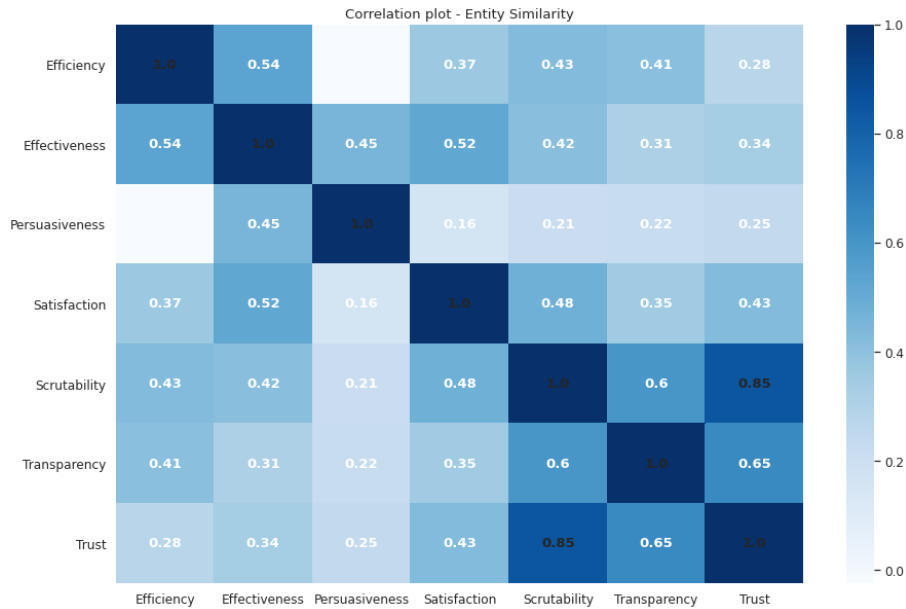
Figure 7.5: Results on explanation through entity relatness as proposed by Ripple-Net[97]

### 7.3.3.6 Similarity

Figure 7.6 depicts visualization of the results in relation to each explanation goal, hereby a stacked visualization of the Likert scores, in addition to a correlation matrix.



(a) Stacked visualisation of the Likert scores

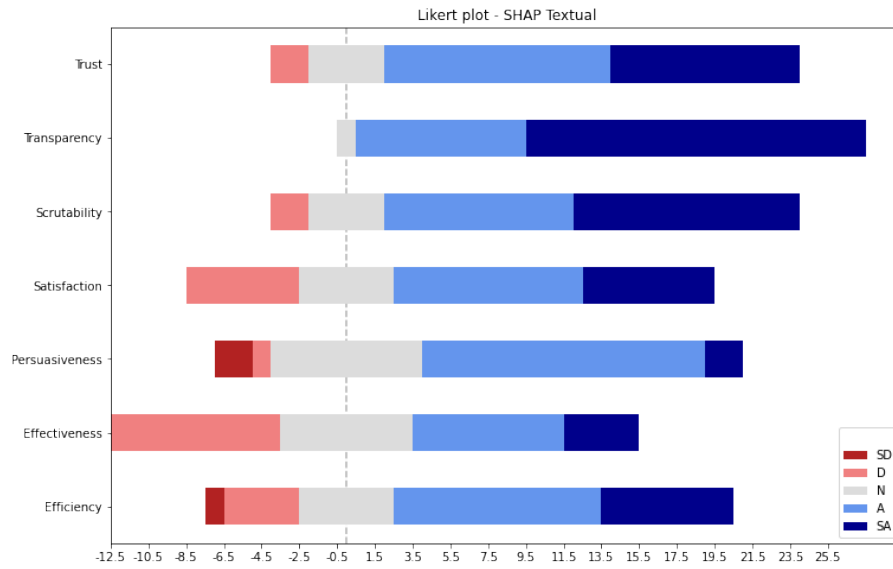


(b) Correlations related to each goal

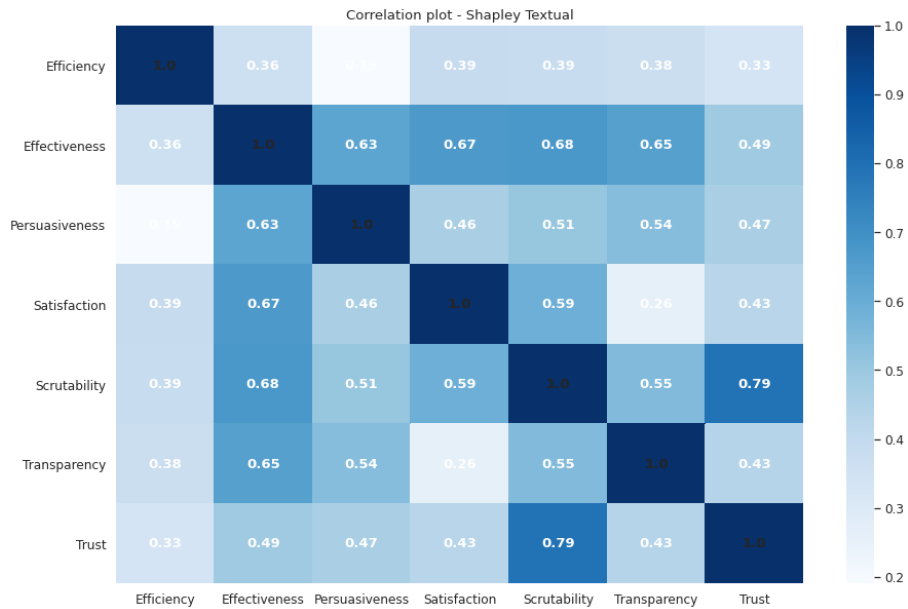
Figure 7.6: Results on explanation through highlighting category and similarity to historic interactions.

### 7.3.3.7 Textual SHAP

Figure 7.7 depicts visualization of the results in relation to each explanation goal, hereby a stacked visualization of the Likert scores, in addition to a correlation matrix.



(a) Stacked visualisation of the Likert scores

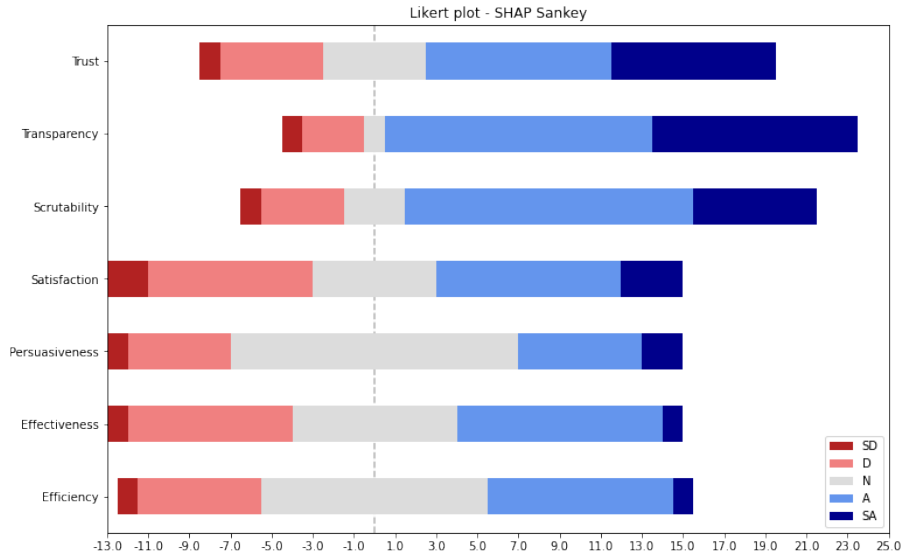


(b) Correlations related to each goal

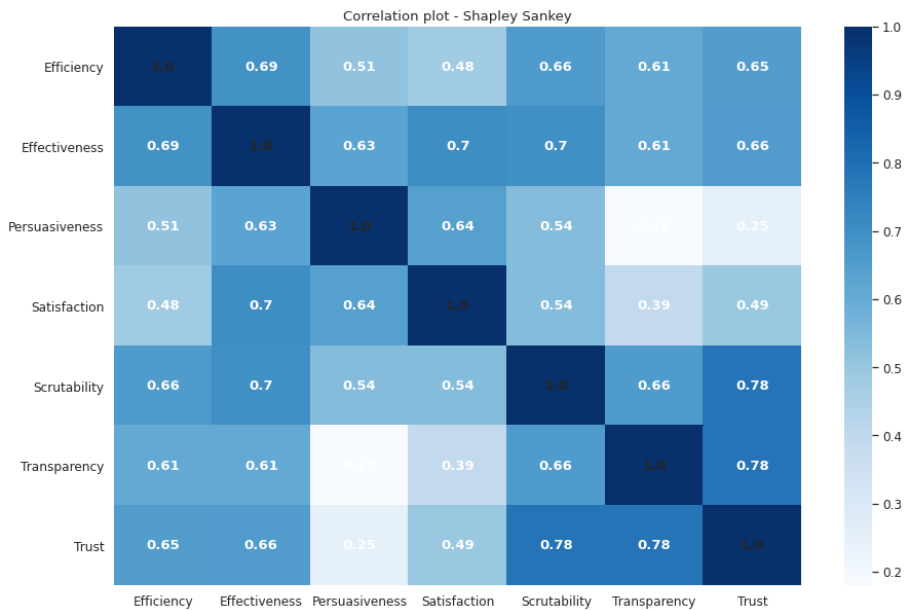
Figure 7.7: Results on explanation through textual feature highlighting of Shapley values.

7.3.3.8 Visual SHAP

Figure 7.8 depicts visualization of the results in relation to each explanation goal, hereby a stacked visualization of the Likert scores, in addition to a correlation matrix.



(a) Stacked visualisation of the Likert scores



(b) Correlations related to each goal

Figure 7.8: Qualitative results on explanation through visual feature highlighting of Shapley values.

	Effe.	Effi.	Persu.	Satis.	Scruta.	Trans.	Trust
<b>Statement</b>	2.43	2.71	3.21	3.11	2.40	1.93	2.50
<b>Category</b>	2.93	3.43	2.82	3.21	3.04	3.21	3.07
<b>Profile</b>	3.61	3.57	3.43	<b>3.93</b>	3.64	3.79	3.5
<b>Snippet</b>	3.14	2.93	3.43	3.11	2.39	2.39	2.89
<b>Similarity</b>	<b>3.86</b>	<b>3.68</b>	<b>3.68</b>	3.82	3.68	4.39	<b>4.07</b>
<b>Ripplenet</b>	2.96	3.21	3.25	3.46	3.11	3.96	3.07
<b>ENSUS Text</b>	3.25	<b>3.68</b>	3.50	3.64	<b>4.15</b>	<b>4.60</b>	<b>4.07</b>
<b>ENSUS Visual</b>	3.07	3.11	3.10	3.10	3.71	4.00	3.64

Table 7.6: Mean evaluation scores with respect to each explanation and evaluation goal. The highest score(s) for each respective goal is marked in bold. Here the Likert values are numbered, with 1 corresponding with *strongly disagree* and 5 corresponding with *strongly agree*.

### 7.3.4 Observations

Table 7.6 provides an overview of the mean evaluation scores for the proposed explanations and baselines with respect to each explanation goal.

#### 7.3.4.1 Efficiency of Explanations

We recall that efficiency in explainable news recommendation is related to how explanations are potentially aiding users in becoming more efficient in navigating the article space. Measuring efficiency through a user survey is difficult compared to that of on-line evaluation tools in which efficiency is inferred from the decision time as performed in [28]. Instead, subjects were asked to assess to what degree they felt the explanation would make them more effective in reading news. The results show that clearly, highlighting a relationship between the recommended item contents and historic interactions as with *profile* and *similarity* have a positive effect on perceived efficiency, as compared to supplying additional content information. An interesting observation is the performance of *rippenet* compared to that of *similarity*. Since *rippenet* is focused on a single entity for inferring relatedness, it seems that users prefer a more general relatedness as with *similarity*, where the relatedness between the articles is considered as opposed to a single named entity.

Additionally, as visualized in figure 7.3 (a), positive ratings on efficiency correlate highly with that of effectiveness. Since this is only evident in the *profile*, *rippenet* and *snippet* we believe this might be related to how the explanations provide additional information about the respective article itself, further allowing users to make informed decisions before even clicking the article.

#### 7.3.4.2 Effectiveness and Persuasiveness

Effectiveness is to what extent the explanation is able to aid users in assessing the quality of the recommended item and make more informed decisions. The results indicate that subjects preferred the *similarity* and *profile* based explanations. Our intuition.



In terms of persuasiveness we observe that no explanation notably outperforms the others, whereas all explanations but *category* are perceived as somewhat persuasive. An interesting observation is that the *snippet* performed equally as well on persuasiveness as *profile*, suggesting that subjects find elaborated background data compelling.

All in all the results show that justifications as explanations incorporating domain specific data or coupled with a relationship between knowledge objects outperforms justifications solely based on background data in terms of effectiveness

Furthermore the persuasiveness of *statement* compared to i.e. *category* highlighting suggest that directly addressing the user in the explanation has an effect on the perceived persuasiveness. We theorize that this is caused by the explanation creating a *personalized feel*, suggesting that simply adjusting the language of textual explanation templates to accommodate a perceived personalization can increase the persuasiveness of recommendations.

### 7.3.4.3 Satisfaction

The evaluation goal of *satisfaction* is meant to evaluate the subjects overall satisfaction with the explanation interface. A surprising observation is that no explanation was evaluated to prove negative satisfaction. In respect to the median scores, the novel *statement* scored barely above "Not Sure". To our surprise, the *Visual* explanation scored below the statement. We theorize that this might be related to how users have little reference in terms of visual explanation styles, as most styles that are utilized in commercial applications are textual. Section 8.2.5 will further elaborate upon the textual versus visual relationship.

Additionally, a further reason to why the simplistic *statement*, *snippet* and *category* did not perform negatively can be related to how these styles nowadays can be found on many popular e-commerce platforms. Thus, users are probably acquainted to rely on this kind of simplistic information in their decision making process.

Interestingly, there seem to be a relationship between the explanations that had a *personalized* appeal, in that they convey a conceptual model that references the user directly, since *profile*, *similarity* and *ENSUS Text* performed best in terms of satisfaction. Surprisingly, the *profile* explanation scored well above *ENSUS Text*. We believe this also might be affiliated with how additional background information is utilized for justifying the recommendations.

As demonstrated by previous work [90, 14, 28], utilizing domain specific background information in explanations improve effectiveness and satisfaction with the explanations, as tangible explanation concepts that require less cognitive load are preferred by the user.

### 7.3.4.4 Perceived Transparency

The perceived transparency of the explanations are of great importance for the contributions of this thesis. Our initial assumptions was that descriptive explanations that are largely concerned with explaining the underlying model will provide substantially

better perceived transparency as compared to descriptions. As shown in table 7.6 the *ENSUS Text* explanation outperforms the baseline descriptions in terms of perceived transparency. Figure 7.7 show that the subjects unanimously agree that the *SHAP Text* explanation performs best in terms of transparency.

#### 7.3.4.5 On Trust

We initially expected to discuss the observations of trust with respect to perceived transparency, as we expected highly correlated results in regards to transparency and trust. However, we are very surprised to find that a justificatory explanation, namely *similarity*, performed just as well as the descriptive *SHAP textual* in terms of users perceived trust in the system. We assume that the personalized appeal, in addition to the familiar *content-based* approach for justification present in *similarity* contributed to this. Compared to transparency, we also assumed that the *SHAP textual* would perform better, but we believe this might be caused by the explanation not having the same elaborate personalized appeal as the *similarity*, in which users are justified through "this was recommended because of what you read earlier", which is a concept of explanation that is familiar and easy to settle with, as compared to the feature values which in turn can seem more abstract in their nature.

Furthermore we are pleased to see that the users reacted negatively to the simple highlighting of background information through *snippet* and, especially *statement*.

#### 7.3.4.6 Relationships Between Variables

Furthermore we are pleased to observe correlations between scrutability and transparency in both descriptive explanations, namely *SHAP Textual* and *SHAP Sankey*, and we theorize that this relates to how the users understood the relationship between their characteristics, and how they affected the recommendations.

Suprisingly, the *similarity* explanation performed best in terms of correlation between scrutability and transparency. Again, this might be a coincidence, but an educated guess is that the personalization aspects of the explanation while highlighting relevant content that might influence the recommendation is affecting how the user considers whether or not the highlighted information truly will affect the recommendation.

Furthermore, we cannot determine any clear trends that are shared among similar explanations or the opposite. This might be related to how we plot or determine the correlations, but we believe also that the number of participants is too low for us to make any assumptions based on minor similarities shared among the explanations.

## 7.4 Qualitative Evaluation

This section covers the qualitative experiments conducted. First, we provide and analyze an example learned by the model. Second, we inspect the Shapley values generated by the learned model in order to assess if they are usable. Finally, we inspect the latent dimensions generated by BERT, explained in chapter 6.3.

### 7.4.1 Qualitative Evaluation of ENSUS

Following recent work [59], we present and analyze an example learned by to model. This will provide a better intuition for the generated explanations of ensus. A textual explanation and a visual explanation are presented to illustrate how the model can be used to generate explanations.

The first example is a textual explanation generated by the learned model:

- Your interest in "newsopinion" contributed 42% to this recommendation, while your interest in "politics" contributed 12%.

This example show a user who have read articles about newsopinion, politics, budget and others. From the Shapley values, we calculate how much each topic in the user profile contributed towards the recommendation and use this as an explanation.

The second example, illustrated in figure 7.9, show a Sankey diagram over the Shapley values.

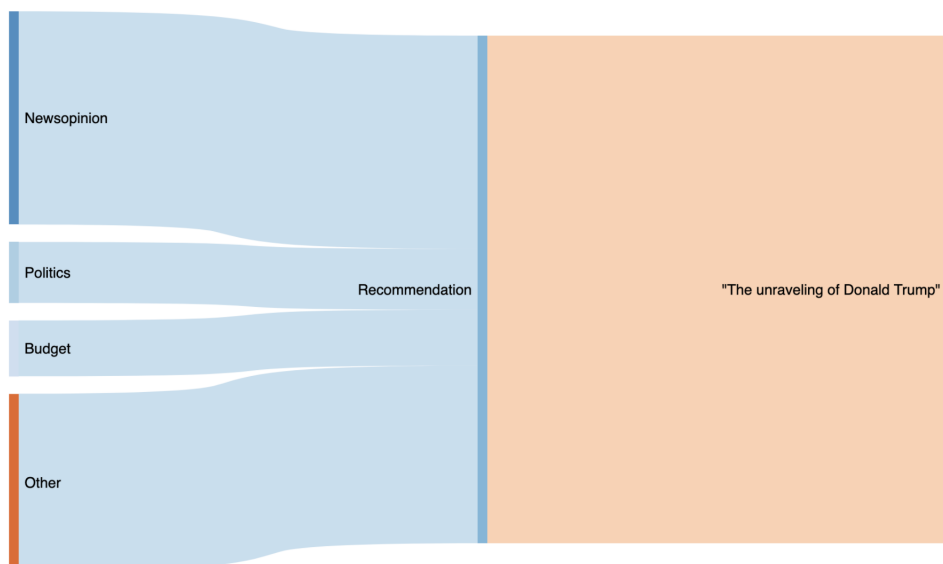


Figure 7.9: Sankey diagram over the shapley values where the article itself is removed from the left siden.

## 7.4.2 Inspecting the Shapley Values

Figure 7.10 show the Shapley value for the top 5 recommendations for 30 randomly sampled users resulting in a total of 150 line plots. The 39 input features are on the x-axis and the corresponding Shapley value is on the y-axis. The first 30 units on the x-axis are article IDs from the user’s click history. The units from 30 to 36 are the user profile, number 37 is the article ID predicted (e.g. the  $n$ th recommended article) and the two last units are the article’s category and subcategory.

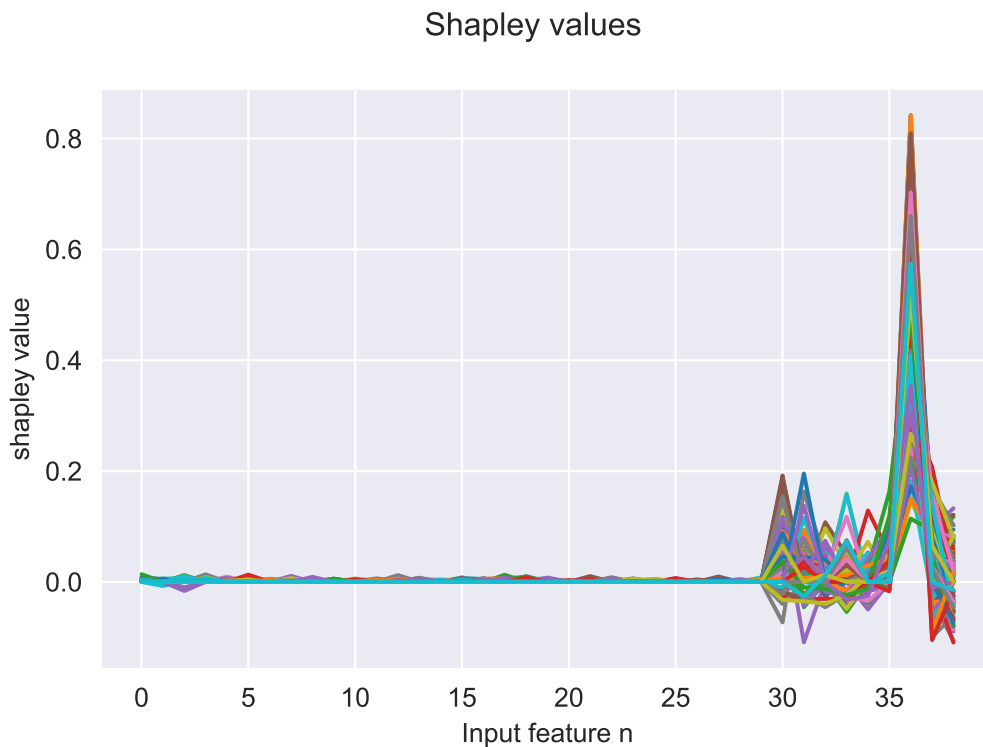


Figure 7.10: The shapley values for top 5 recommendations for 30 randomly sampled users.

We see that the user’s click history have a limited importance in the model except for a few clicks where the contribution is either positive or negative. Furthermore, we see that the user profile and the article’s category and sub-category influences the prediction. However, the predicted article have substantially more importance to the predictions.

Figure 7.11 depicts the Shapley value plot where the maximum amount of articles in the click history is set to 10 with a total amount of 19 input features to the model. We see that the importance of click history has not improved compared to the case where the length of the click history where 30, depicted in figure 7.10.

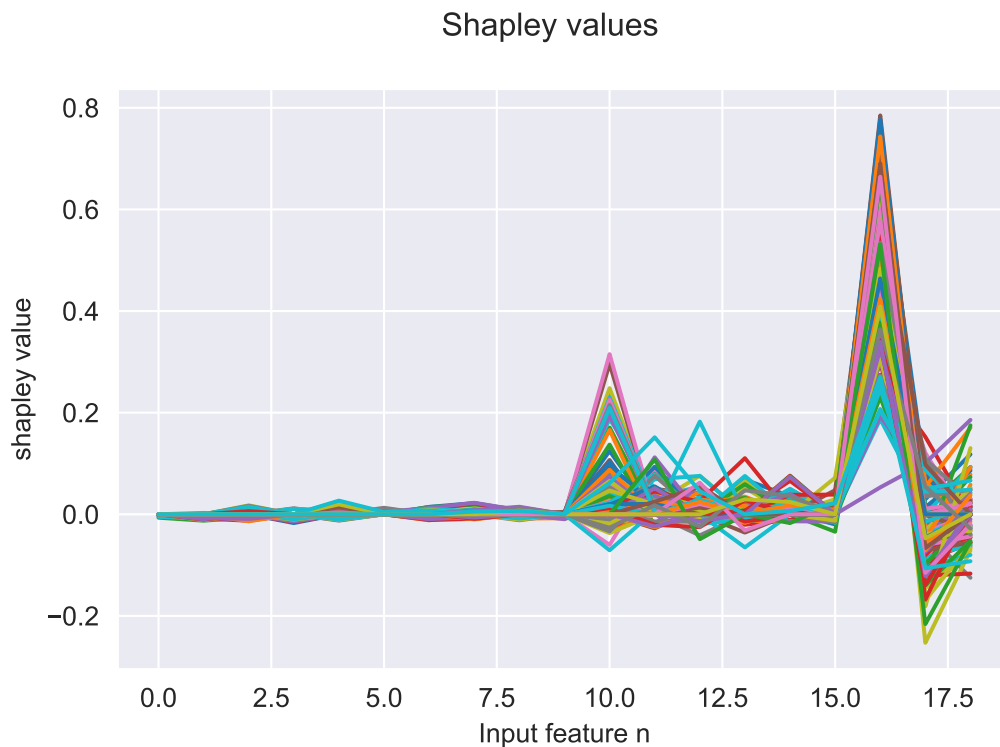


Figure 7.11: The shapley values for top 5 recommendations for 30 randomly sampled users where the maximum amount of articles in the click history is set to 10.

In other words, ENSUS is not capable to include click history in its predictions. As a result, it is not possible to provide explanations that refer to previously read articles. Further work has to be made in order to find an architecture that can include click history.

Even though ENSUS fails at click history, the figures show the user profile and article topics can be used to provide explanations.

### 7.4.3 Visualization of Latent Dimensions

The highly dimensional BERT embeddings are in their raw form intangible to humans. However, we can allow for a visual interpretation of the relatedness between news-articles by reducing the dimensionality through projecting the embeddings onto a two-dimensional plane.

To visualize the relatedness between the BERT representations we utilize a technique known as t-sne [95] for projecting these highly dimensional embeddings into two dimensions.

We visualize the MIND embeddings in figure 7.12. As depicted by the legends, each color is representative of a specific factor, in this case a news category labeled by the article author (namely "label1"). Based on the clustering of factors we observe that the clustering phenomenon is consistent with affiliating factors.

Furthermore we observe that the right uppermost part of the cluster is dominated by

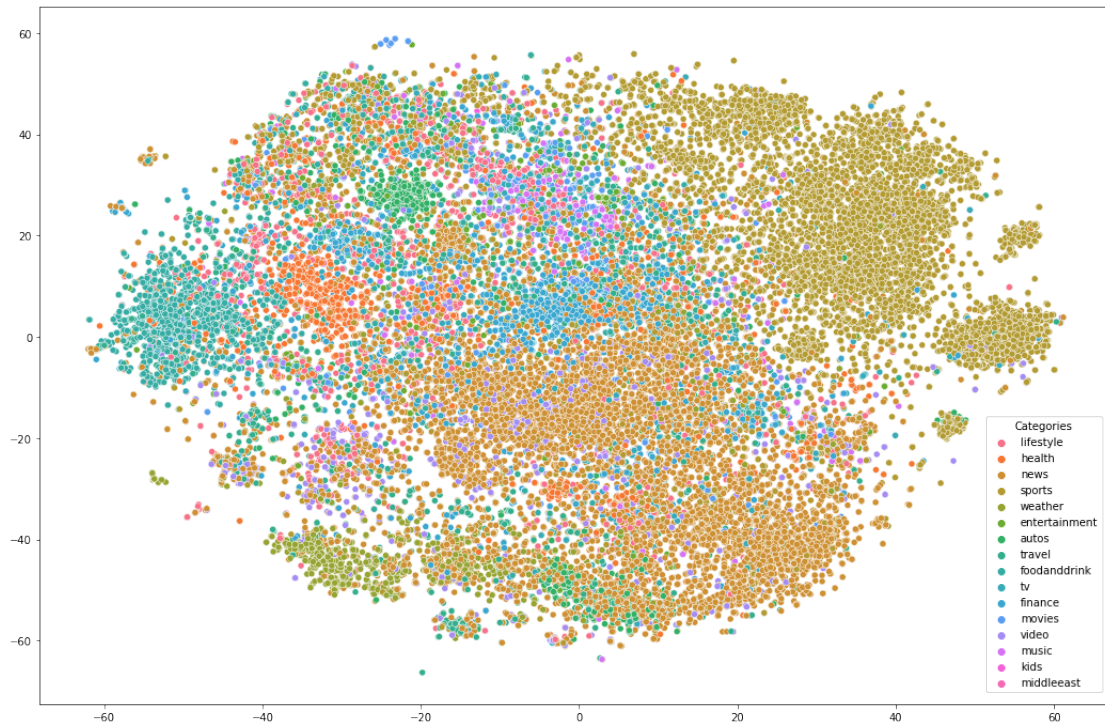


Figure 7.12: t-SNE visualization of embeddings from MIND article abstracts, colored according to article category

articles labeled "sport", in which several individual sub-clusters can be observed. By introducing the subcategories in place of news categories (namely "label2") we get a new plot as depicted in figure 7.13, and we observe a consistency in the clustering phenomenon as articles affiliated with different sport disciplines are contained as sub-clusters within the "sport" cluster.

This demonstrates that the embedding model used is able to discover different factors from the article abstracts in the dataset and assign embeddings accordingly. As mentioned in section 6.3.2, we'll only be considering the entity similarity on the MIND dataset. We performed experiments on the Adressa dataset as well, but received unnaturally high values of accuracy when using the Norwegian notram BERT based transformer model. As is covered by [79], the reasoning for the high accuracy is based on how BERT based transformer models in themselves are not trained for sentence similarity directly. Luckily, Reimers et al.[79]



Figure 7.13: t-SNE visualization of embeddings from MIND article abstracts, colored according to article subcategory

### 7.4.3.1 Model Fidelity

To qualitatively evaluate the explanations based on entity relatedness we adopt an approach from Peake et al.[76], who generalized the concept of *model fidelity* based on earlier research by Abdollahi et al.[2].

As described in section 4.4.3, model fidelity is defined as the percentage of explainable items in the recommended items:

$$\text{Model Fidelity} = \frac{|\text{explainable items} \cap \text{recommended items}|}{|\text{recommended items}|} \quad (7.1)$$

In the context of the entity relatedness model, an item is deemed explainable if there exist a different item in the user history where the cosine similarity between the embeddings are above the threshold of 0.6, thus allowing for an explanation on the form "*This item article is about [category], similar to [historic\_article]*"

The basis of our model fidelity is a sample 30 users sampled randomly from the MIND dataset. As shown in table 7.7, the average model fidelity is around right above 26%, suggesting that 1 in 4 articles can be explained using the justification approach on entity relatedness.

	MIND
@0.6	0.267

Table 7.7: Model fidelity at cosine similarity threshold of 0.6

## 7.5 Evaluating Scrutability

The user profile is an important component for the proposed method as it substantiates the explanation goal scrutability, in that the user profile keeps a record of the assumed interests of a user. Scrutability is achieved through allowing users to alter their own user profile by adding or removing categories of preference.

The user profile offer the user to give feedback to the recommendation engine. For example, consider a user on Spotify with children. The user plays children music for her children and, after some time, the user will be recommended children music. A user profile would enable the user to remove the children genre from her user profile and then consequently remove children music recommendations.

However, scrutability is only enabled if the user profile have a significant influence on the predictions. If a user make changes to the user profile, the recommendations have to change. Consequently, topics from the user profile needs to be present in the top  $k$  recommendations. For example, if a user likes golf and baseball, then golf and baseball should be present in the top 10 recommendations.

For these reasons several neural architectures were developed in order to find an architecture that facilitates the use of a user profile that meet the requirements discussed above. Each experimental architecture is trained for 10 epochs with early stopping on the two datasets Adressa and MIND. Table 7.8 presents the different architecture used in this experiment. Architecture 1, 2, 3, and 4 are baselines. Architecture 5 and 6 are architectures developed in addition to the proposed method in order to find an architecture that best suited goal.

	Architecture	Input features
Architecture 1	NCF	User ID and Article ID
Architecture 2	NeuMF	UserID, article ID, user profile
Architecture 3	NeuMF	User click history, article ID, article category, article subcategory, user profile.
Architecture 4	WideDeep	User click history, article ID, article category, article subcategory, user profile.
Architecture 5	Session-based	User click history, user profile, article ID and article features (LSTM over clickhistory)
Architecture 6	Session-based	User click history, user profile, article ID and article features (two tower LSTM over profile and history)

Table 7.8: Architectures used to evaluate the presence of the user profile.

To evaluate the presence of the user profile in the top  $k$  recommendations we use  $\text{Count}@k$ , defined in equation 7.2.  $\text{Count}@k$  is the fraction of number of times each topic in the user profile is present in the top  $k$  recommendations, denoted  $C(k, u)$ , and the number of users in the test set,  $N(u)$ . Figure 7.14 illustrates how  $C(k, u)$  is calcu-



lated. In the example,  $C(k, u) = 3$ .

$$\text{Count}@k = \frac{C(k, u)}{N(u)} \quad (7.2)$$

$\text{NDCG}@k$  is used in addition to  $\text{Count}@k$  where the highest topic in the recommendation list is used to calculate the  $\text{DCG}$ .  $k$  ranges over  $\{5, 10\}$  in the experiments.

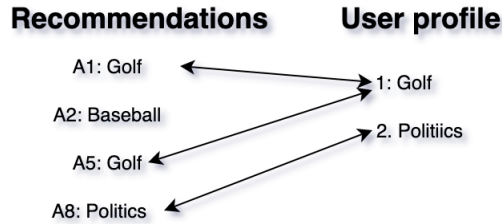


Figure 7.14:  $C(k, u)$

To evaluate the scrutability of ENSUS, we perform an experiment where randomly sample 10 users from the MIND dataset, and create 10 recommendations for each user. After calculating the  $\text{Count}@k$  on the recommendations, we changed the entire user profile by randomly sample news topics from the MIND dataset. Finally, the new  $\text{Count}@k$  was calculated on the newly created recommendations.

Note that the  $\text{Count}@k$  values before the scrutinization of the user profile is different from the values reported in table 7.9 since the sample space was limited to 10 users.

### 7.5.1 Scrutability Results

The presence of the user profile from the architectural experiments are reported in table 7.9 and table 7.10 in terms of  $\text{Count}@K$  and  $\text{NDCG}@K$ . We see that the proposed model outperforms the other architectures on the MIND dataset. However, on the Adressa dataset, we see that both version of NeuMF performs slightly better in terms of  $\text{Count}@k$ . The reason might be that the quality on the categories are much better on the MIND dataset. The Adressa dataset have 0.0035 unique categories per article while MIND has 0.0074 unique sub-categories per article. This is clear from the results where we see that the  $\text{Count}@k$  and  $\text{nDCG}@k$  are higher on the Adressa dataset. Since ENSUS is specifically designed to include categories in the prediction, this will have an impact on the results. The results verifies that putting a LSTM over the user profile together with the article category increases the importance of the user profile for the top  $K$  recommendations.

An interesting observation is that both Architecture 5 and 6 are outperformed by all models except for the NCF implementation. We thought that using an LSTM to model a user's click history would be more logical. The reason may be that the input data is not strictly modelled to conform to RNNs but rather as a user-article tuple. Consequently, for some data instances, the model may see only one true instance of a click history and

4 negative instances of the click history (as we sample 4 negative articles) if a user has a total of two clicks.

Methods	Count@5	nDCG@5	Count@10	nDCG@10
Architecture 1	1.42	0.60	2.52	0.64
Architecture 2	1.48	0.59	2.82	0.64
Architecture 3	1.45	0.65	2.54	0.69
Architecture 4	1.51	0.71	2.91	0.78
Architecture 5	1.38	0.55	2.4	0.60
Architecture 6	1.40	0.58	2.56	0.62
Proposed Model	<b>2.11</b>	<b>0.78</b>	<b>3.45</b>	<b>0.80</b>

Table 7.9: The performance of different methods on the MIND dataset

Methods	Count@5	nDCG@5	Count@10	nDCG@10
Architecture 1	3.21	0.84	6.41	0.84
Architecture 2	3.24	0.86	6.43	0.85
Architecture 3	<b>3.45</b>	<b>0.88</b>	<b>6.56</b>	<b>0.88</b>
Architecture 4	3.37	0.85	6.47	0.86
Architecture 5	3.23	0.85	6.31	0.85
Architecture 6	3.23	0.85	6.30	0.85
Proposed Model	3.27	0.85	6.45	0.86

Table 7.10: The performance of different methods on the Adressa dataset

Table 7.11 reports the experiments related to scrutability. The left column reports the Count@k for the case when the user profile is the original profile. The right column reports Count@k where the user profile is changed to another random user profile.

Compared to table 7.9 we have several observations. First, we see that scrutinized ENSUS outperform the non-scrutinized versions of NCF, NeuMF, and both of the session-based recommenders. Second, we see that Count@k drops significantly for the scrutinized case. This illustrates the trade off between the collaborative filtering effect and the feature importance push on the user profile. While it is possible to scrutinize to some extent, it is not possible to completely remove the collaborative effect.

	Original	Scrutinized
Count@10	3.49	2.56
Count@5	2.17	1.49

Table 7.11: Left column report results in terms of Count@k with the original user profile. In the right "scrutinized" column, the user profile consists of randomly sampled topics.

## 7.6 Performance Evaluation

To evaluate the performance of the proposed recommender system, we compare it against the following baselines:

- Popularity-based recommendations
- ALS [50]
- NeuMF [37]
- WideDeep [21]

To measure the recommendation performance in terms of accuracy we use Hit Rate (HR) and Normalized Discounted Cumulative Gain (NDCG).

For each user  $u$ , we sort the items in descending order according to the predicted probability of a user clicking the item  $i$ . An item is relevant in the test set if the item is the last clicked item in the user's click history. Hit Rate is the fraction of number of times the relevant item is retrieved among the top- $k$  ranked items, denoted  $N(k, u)$  and the number of users in the test set,  $N(u)$ :

$$HR@k = \frac{N(k, u)}{N(u)} \quad (7.3)$$

NDCG measure the ranking quality and reward relevant items that appear higher in the top- $k$  recommendations. NDCG is defined as follows:

$$NDCG = \frac{DCG_{pos}}{IDCG} \quad (7.4)$$

where  $IDCG$  is the ideal ranking, e.g. that the relevant item appears first in the ranked list, and  $DCG_p$  is defined as:

$$DCG = \sum_{pos=1}^k \frac{relevance}{\ln(pos + 1)} \quad (7.5)$$

### 7.6.1 Performance Results

The performance of the proposed model and the baselines are reported in table 7.12 and table 7.13 in terms of Hit@ $k$  and NDCG@ $k$  on the MIND dataset and the Adressa dataset respectively.  $k$  ranges over  $\{5, 10\}$ . NeuMF and WideDeep are reported with and without the same feature set as the proposed ENSUS model.

From table 7.12 and 7.13 we have several observations. First, we see that the methods based on neural networks outperform the popularity based method and the ALS. This is because neural networks can learn better news and user representations than traditional matrix factorization methods. Second, ENSUS has better performance compared to the baselines on the MIND dataset. However, the results are not that clear on the Adressa

dataset. The reason might be that the quality on the categories are better on the MIND dataset. As mentioned in the aforementioned section, Adressa have 0.0035 unique categories per article while MIND has 0.0074 unique sub-categories per article. Since ENSUS is specifically designed to include categories in the prediction, this will have an impact on the results. The results validates that news topics are useful for news recommendations and that ENSUS can exploit the topic information.

Methods	HIT@5	nDCG@5	HIT@10	nDCG@10
Popularity based	0.03	-	0.05	-
ALS	0.07	-	0.10	-
NCF	0.12	0.07	0.14	0.08
NeuMF	0.52	0.37	0.66	0.44
NeuMF with features	0.49	0.35	0.69	0.41
Wide&Deep with features	0.60	0.44	0.73	0.48
ENSUS	<b>0.64</b>	<b>0.50</b>	<b>0.78</b>	<b>0.54</b>

Table 7.12: The performance of different methods on the MIND dataset

Methods	HIT@5	nDCG@5	HIT@10	nDCG@10
Popularity based	0.01	-	0.02	-
ALS	0.08	-	0.16	-
NCF	0.22	0.18	0.36	0.25
NeuMF	0.23	<b>0.19</b>	0.37	<b>0.26</b>
NeuMF with features	0.18	0.12	<b>0.43</b>	0.23
Wide&Deep with features	<b>0.25</b>	0.13	0.30	0.25
ENSUS	0.24	0.15	0.42	0.25

Table 7.13: The performance of different methods on the Addressa dataset

## 7.7 Observations

### 7.7.1 Presentation Style

In regards to preferences on presentation styles our quantitative evaluation results demonstrate that users prefer the template-based textual presentations of feature relevancy. Although we thought the *ENSUS visual* had a visual appeal, we are not surprised that the visual presentation scored notably worse than the template-based textual visualization. However, we did not expect the visualization to score worse than the *statement* in terms of satisfaction. We theorize that this might be related to how a visual explanation very far from what little explanations users have experienced, as they are not that common in commercial applications. However, we believe this might also be a bias due to the visualization being the last survey question, and it being the only visualization amongst the eight explanations.

## 7.7.2 Descriptions and Justifications

The broad categorization of explanations as either descriptions and justifications as described in [45] does not provide any assessment to whether the one is expected to be a more efficient explanation than the other. However, due to the Shap based descriptions being first and foremost concerned with explaining the underlying model in a tangible manner, we expected the descriptive explanations to receive high scores on *transparency* and *trust*, as these goals are mainly concerned with whether or not the user has a perceived understanding of *how* the system works through transparency, and thereby has an increased trust the system. Naturally we would therefore expect these goals to correlate greatly in the descriptive explanations.

Based on the results we observe that the SHAP Textual explanation performs well above all other explanations in terms of scrutability and transparency. In terms of trust and efficiency, the SHAP Textual is tied with the Similarity approach. On the contrary, both SHAP models performs notably worse in terms of effectiveness and satisfaction, where surprisingly the SHAP Visual performs barely worse than the recommendation statement explanation in terms of satisfaction.

This substantiates our initial expectations, that the descriptive explanations will perform very well in terms of transparency, trust and scrutability. But does not necessarily outperform descriptive explanations in terms of persuasiveness and satisfaction

## 7.7.3 Explanation Efficiency

A major limitation with surrogate models such as LIME and SHAP is that they are slow. The time complexity of SHAP grows with the number of features as it performs permutations over the input feature space.

We performed the calculation of the Shapley values with the maximum amount of input features for our model, e.g. 39 features. On average, it took 0.37 seconds to calculate the Shapley values. The main reason for the very short calculation time is that we provide Shapley values for one single recommendation at a time. In addition, the feature space is relatively small compared to models that may take hundreds of input features.



# Discussion and Further Work

The following chapter presents a discussion and evaluation of the goal and RQs in section 8.1. Section 8.2 discusses limitations and further work.

## 8.1 Conclusion

The objective of this thesis was to *Explore how state-of-the-art descriptive explanations compare to justifications in regards to providing trust, transparency and scrutability for a neural news recommender system*. Four RQs were formulated to reach the goal..

**RQ1** *What is explainability in the domain of recommender systems and what is the state of the art in providing explanations alongside recommendations?*

To answer this question, we provided a brief taxonomy on explainable recommender systems in chapter 3. We defined explainability as an interface between humans and the prediction model. Furthermore, an explanation can be either descriptive, which reveal the actual mechanisms that generated the recommendations, or a justification, which provides a conceptual model that may differ from the underlying recommendation model.

Specific to explainability on recommender systems are the three levels of explanations: transparency, contextualization and self-actualization. Transparency serves as the foundation for explainability and provides insight through descriptive information such as describing the raw data used in the recommendations. The second layer, contextualization, combines historic interactions with items and thereby offers contextualization. The final layer, self-actualization, allows user to understand their own unique tastes and preferences.

Following recent work, we presented six different methods for model-intrinsic approaches, namely; matrix factorization models, topic modeling, graph-based models, deep learning, knowledge graph-based and rule mining. Model agnostic methods for explaining recommendations were LIME and SHAP.

Finally, we propose to classify existing research in explainable recommender systems with respect to three orthogonal dimensions: the information source (what information is being used in the recommendations), the presentation style (how is this information used) and the presentation style (how is the information presented to the user).

**RQ2 *How does the explanations in the proposed method compare to the state-of-the-art explainable approaches in the literature?***

A user survey was conducted in order to quantitatively compare the proposed ENSUS model against RippleNet. The results show that the textual ENSUS model outperforms RippleNet on all seven goals, e.g. effectiveness, efficiency, persuasiveness, satisfaction, scrutability, transparency and trust.

In addition to RippleNet, we compared ENSUS against simple textual justification, highlighting category, category profile, abstract snippet and similarity. The results show that ENSUS outperforms the different explanations on efficiency, scrutability, transparency and trust. However, the similarity explanation has best performance on effectiveness, efficiency and persuasiveness.

**RQ3 *How does state-of-the-art descriptive explanations compare to justifications in terms of transparency and trust?***

In the context of RQ3, both explanations generated by ENSUS is representative for the state-of-the-art in descriptive explanations. They incorporate a model-agnostic approach to explainability, describing the neural classification efforts through Shapley values. This approach is highly concentrated on explaining the underlying model, bringing transparency through detailing the importance of tangible features such as article categories for the user. To answer this research question, we evaluated the explanations through a detailed user survey. Furthermore, we expected this approach to outperform the justifications in terms of bringing transparency to the recommendation process, and based on this we expected the concept of transparency to be highly correlated with perceived trust. As expected, the

With this in mind, we conclude that indeed, state-of-the-art descriptive explanations do outperform justifications in terms of perceived transparency to the recommendation process. However, the perceived transparency does not necessarily foster trust in the recommendation process. Surprisingly, our proposed *similarity* approach, which fully omits the black-box classification of the ENSUS recommender system, performed equally as well in terms of trust in the recommendation process.

This demonstrates that implementing complex descriptive methods for explanation will indeed foster transparency, but highlighting the relationship between the users actions and the recommendations will possibly result in an equal amount of trust in the recommendation process, given that the highlighted aspects are warranted for. However, these relationships are not guaranteed to be present, as fully omitting the black-box means the explanations are based on an assumption, whereas we know that recommender systems implementing collaborative measures can recommend items that are totally unrelated to a users previous interactions. With a model fidelity of one in four, it is debatable to whether or not such an approach is reliable in a commercial setting.



#### **RQ4 *What are the advantages and disadvantages of the proposed methods?***

Qualitative experiments were performed to highlight the advantages and disadvantages of ENSUS. By inspecting the values of the generated Shapley values the results show that the ENSUS model was not able to tell the user what articles in the click history contributed towards the recommendations except for a few specific cases. That being said, the plot validates that news topics are useful for news recommendations and that ENSUS can exploit the topic information to provide explanations.

Contrary to [26], our experiments related to efficiency show that the time to calculate the Shapley values was not of any concern (e.g. 0.37 seconds). Since [26] has not provided any code for this particular experiments, we were not able to replicate their results.

## **8.2 Further Work**

### **8.2.1 Finetuning Hyperparameters and Model Architecture**

Experiments in this thesis did not include finetuning of the hyperparameters. Consequently, results may be different if we would have used more time on finetuning the hyperparameters. Furthermore, we had issues with using the published code on several methods mentioned in this thesis. This limited our experiments.

### **8.2.2 Self-Actualization**

As proposed by Sullivan et al.[90] and as mentioned in section 3.3.3, the third layer of explanation foster self-actualization through supporting epistemic goals — further allowing users a direct control over which goals they wish to actualize. Neither our proposed explanatory model for feature relevance nor historic similarity support this layer of transparency. Our preliminary testing on self-actualization showed little to no results in improving the accuracy of recommendations.

However, self-actualization is not restricted to entertainment related commercial recommender systems or those of e-commerce, and can be related to many classification tasks. Imagine a future home owner applying for a mortgage loan. After entering details on income, age, education etc. into the algorithm, the user is simply presented with a denial — a result of a complex computation beyond the users comprehensiveness. This does not foster self-actualization. However, presenting the user with an explanation on the form "*You would have received the loan if you made 5.000\$ more*", will further promote self actualization. This is related to *counterfactual* explanations[96], where simply providing a "what if" scenario for the model input can function as a explanation in itself.

Furthermore, allowing users to explore different "what if" scenarios can further foster self-actualization, in a similar manner to how children learn through counterfactual examples[68]. Further work on fostering self-actualization in news recommender systems could therefore attempt to include a counterfactual explanatory model to compliment

those proposed in this thesis, allowing users to explore "variations" of their own user-profile through meddling with features of relevance.

### **8.2.3 Efficiency**

Since our experiments on efficiency significantly diverged to results from [26], further work should be done to validate our results. Note that the results from [26] is on LIME. Since SHAP is built on top of LIME, we would expect similar results. Furthermore, we could not find any research on the calculation time on SHAP or any quantitative results showing that SHAP is slow.

### **8.2.4 Improving ENSUS Architecture**

As discussed in section 8.1, ENSUS was not able to use the information from the click history. Consequently, it was not possible to include the click history into the explanations. Improvements on the model architecture should be made so that the click history can be included.

Despite the good performance of ENSUS, we acknowledge that the recommender component is too simplistic compared to the plain recommender systems in the state-of-the-art. Consequently, we should improve the recommender component of ENSUS to include attention-mechanisms and convert the entire component to a session-based recommender.

### **8.2.5 The Assumption that SHAP is Reliable**

This thesis makes the assumption that SHAP is accurate and that its results are correct. The details behind SHAP is very complex and we assume that the SHAP library works as promised without any questions. Further work has to be done in order to validate that our assumptions are correct, and the fact that SHAP is reliable.



# Bibliography

- [1] Kjersti Aas, Martin Jullum and Anders Løland. ‘Explaining individual predictions when features are dependent: More accurate approximations to Shapley values’. In: *arXiv preprint arXiv:1903.10464* (2019).
- [2] Behnoush Abdollahi and Olfa Nasraoui. ‘Using explainability for constrained matrix factorization’. In: *Proceedings of the Eleventh ACM Conference on Recommender Systems*. 2017, pp. 79–83.
- [3] Charu C Aggarwal et al. *Recommender systems*. Vol. 1. Springer, 2016.
- [4] Alejandro Barredo Arrieta et al. ‘Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI’. In: *Information Fusion* 58 (2020), pp. 82–115.
- [5] Dzmitry Bahdanau, Kyunghyun Cho and Yoshua Bengio. ‘Neural machine translation by jointly learning to align and translate’. In: *arXiv preprint arXiv:1409.0473* (2014).
- [6] Krisztian Balog and Filip Radlinski. ‘Measuring Recommendation Explanation Quality: The Conflicting Goals of Explanations’. In: *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2020, pp. 329–338.
- [7] Krisztian Balog, Filip Radlinski and Shushan Arakelyan. ‘Transparent, scrutable and explainable user models for personalized recommendation’. In: *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2019, pp. 265–274.
- [8] Oren Barkan et al. ‘Explainable recommendations via attentive multi-persona collaborative filtering’. In: *Fourteenth ACM Conference on Recommender Systems*. 2020, pp. 468–473.
- [9] Konstantin Bauman, Bing Liu and Alexander Tuzhilin. ‘Aspect based recommendations: Recommending items with the most valuable aspects based on user reviews’. In: *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2017, pp. 717–725.
- [10] Michael A Beam. ‘Automating the news: How personalized news recommender system design choices impact news reception’. In: *Communication Research* 41.8 (2014), pp. 1019–1041.
- [11] Dane Bertram. ‘Likert scales’. In: *Retrieved November 2.10* (2007).
- [12] Mustafa Bilgic and Raymond J Mooney. ‘Explaining recommendations: Satisfaction vs. promotion’. In: *Beyond Personalization Workshop, IUI*. Vol. 5. 2005, p. 153.

- [13] Daniel Billsus and Michael J Pazzani. 'A personal news agent that talks, learns and explains'. In: *Proceedings of the third annual conference on Autonomous Agents*. 1999, pp. 268–275.
- [14] Roi Blanco et al. 'You should read this! let me explain you why: explaining news recommendations to users'. In: *Proceedings of the 21st ACM international conference on Information and knowledge management*. 2012, pp. 1995–1999.
- [15] J. Bobadilla et al. 'Recommender systems survey'. In: *Knowledge-Based Systems* 46 (2013), pp. 109–132. ISSN: 0950-7051. DOI: <https://doi.org/10.1016/j.knsys.2013.03.012>. URL: <https://www.sciencedirect.com/science/article/pii/S0950705113001044>.
- [16] Bruce G Buchanan and Edward H Shortliffe. 'Rule-based expert systems: the MYCIN experiments of the Stanford Heuristic Programming Project'. In: (1984).
- [17] Giuseppe Casalicchio, Christoph Molnar and Bernd Bischl. 'Visualizing the feature importance for black box models'. In: *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer. 2018, pp. 655–670.
- [18] Alexandre Chanson, Nicolas Labroche and Willème Verdeaux. 'Towards Local Post-hoc Recommender Systems Explanations'. In: (2021).
- [19] Hugh Chen, Scott Lundberg and Su-In Lee. 'Explaining models by propagating Shapley values of local components'. In: *Explainable AI in Healthcare and Medicine*. Springer, 2021, pp. 261–270.
- [20] Xu Chen et al. 'Visually explainable recommendation'. In: *arXiv preprint arXiv:1801.10288* (2018).
- [21] Heng-Tze Cheng et al. 'Wide & deep learning for recommender systems'. In: *Proceedings of the 1st workshop on deep learning for recommender systems*. 2016, pp. 7–10.
- [22] Paul Covington, Jay Adams and Emre Sargin. 'Deep neural networks for youtube recommendations'. In: *Proceedings of the 10th ACM conference on recommender systems*. 2016, pp. 191–198.
- [23] Jacob Devlin et al. 'BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding'. In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, June 2019, pp. 4171–4186. DOI: 10.18653/v1/N19-1423. URL: <https://www.aclweb.org/anthology/N19-1423>.
- [24] Fan Du et al. 'EventAction: A visual analytics approach to explainable recommendation for event sequences'. In: *ACM Transactions on Interactive Intelligent Systems (TiiS)* 9.4 (2019), pp. 1–31.
- [25] Leon Festinger. *A theory of cognitive dissonance*. Vol. 2. Stanford university press, 1957.
- [26] Francesco Fusco et al. 'RecoNet: An Interpretable Neural Architecture for Recommender Systems.' In: *IJCAI*. 2019, pp. 2343–2349.
- [27] R Kelly Garrett. 'Echo chambers online?: Politically motivated selective exposure among Internet news users'. In: *Journal of computer-mediated communication* 14.2 (2009), pp. 265–285.

- [28] Fatih Gedikli, Dietmar Jannach and Mouzhi Ge. ‘How should I explain? A comparison of different explanation types for recommender systems’. In: *International Journal of Human-Computer Studies* 72.4 (2014), pp. 367–382.
- [29] Yoav Goldberg and Omer Levy. ‘word2vec Explained: deriving Mikolov et al.’s negative-sampling word-embedding method’. In: *arXiv preprint arXiv:1402.3722* (2014).
- [30] Ian Goodfellow, Yoshua Bengio and Aaron Courville. *Deep Learning*. The MIT Press, 2016. ISBN: 0262035618.
- [31] Jon Atle Gulla et al. ‘The adressa dataset for news recommendation’. In: *Proceedings of the international conference on web intelligence*. 2017, pp. 1042–1048.
- [32] Jon Atle Gulla et al. ‘The Intricacies of Time in News Recommendation.’ In: *UMAP (Extended Proceedings)*. 2016.
- [33] David Gunning. ‘Explainable artificial intelligence (xai)’. In: *Defense Advanced Research Projects Agency (DARPA), nd Web 2.2* ().
- [34] F Maxwell Harper and Joseph A Konstan. ‘The movielens datasets: History and context’. In: *Acm transactions on interactive intelligent systems (tiis)* 5.4 (2015), pp. 1–19.
- [35] William Hart et al. ‘Feeling validated versus being correct: a meta-analysis of selective exposure to information.’ In: *Psychological bulletin* 135.4 (2009), p. 555.
- [36] Chen He, Denis Parra and Katrien Verbert. ‘Interactive recommender systems: A survey of the state of the art and future research challenges and opportunities’. In: *Expert Systems with Applications* 56 (2016), pp. 9–27.
- [37] Xiangnan He et al. ‘Neural collaborative filtering’. In: *Proceedings of the 26th international conference on world wide web*. 2017, pp. 173–182.
- [38] Reinhard Heckel et al. ‘Scalable and interpretable product recommendations via overlapping co-clustering’. In: *2017 IEEE 33rd International Conference on Data Engineering (ICDE)*. IEEE. 2017, pp. 1033–1044.
- [39] Jonathan L Herlocker, Joseph A Konstan and John Riedl. ‘Explaining collaborative filtering recommendations’. In: *Proceedings of the 2000 ACM conference on Computer supported cooperative work*. 2000, pp. 241–250.
- [40] Balázs Hidasi et al. ‘Dlrs 2017: Second workshop on deep learning for recommender systems’. In: *Proceedings of the Eleventh ACM Conference on Recommender Systems*. 2017, pp. 370–371.
- [41] Maartje ter Hoeve et al. ‘Faithfully explaining rankings in a news recommender system’. In: *arXiv preprint arXiv:1805.05447* (2018).
- [42] Liang Hu et al. ‘Interpretable recommendation via attraction modeling: Learning multilevel attractiveness over multimodal movie contents’. In: *IJCAI International Joint Conference on Artificial Intelligence*. 2018.
- [43] Sergey Ioffe and Christian Szegedy. ‘Batch normalization: Accelerating deep network training by reducing internal covariate shift’. In: *International conference on machine learning*. PMLR. 2015, pp. 448–456.
- [44] Folasade Olubusola Isinkaye, YO Folajimi and Bolande Adefowoke Ojokoh. ‘Recommendation systems: Principles, methods and evaluation’. In: *Egyptian Informatics Journal* 16.3 (2015), pp. 261–273.

- [45] J. Riedl J. Vig S. Sen. ‘Tagsplanations: Explaining Recommendations Using Tags’. In: (2009).
- [46] Mozghan Karimi, Dietmar Jannach and Michael Jugovac. ‘News recommender systems—Survey and roads ahead’. In: *Information Processing & Management* 54.6 (2018), pp. 1203–1227.
- [47] Diederik P Kingma and Jimmy Ba. ‘Adam: A method for stochastic optimization’. In: *arXiv preprint arXiv:1412.6980* (2014).
- [48] Bart P Knijnenburg, Saadhika Sivakumar and Daricia Wilkinson. ‘Recommender systems for self-actualization’. In: *Proceedings of the 10th acm conference on recommender systems*. 2016, pp. 11–14.
- [49] Joseph A Konstan et al. ‘Grouplens: Applying collaborative filtering to usenet news’. In: *Communications of the ACM* 40.3 (1997), pp. 77–87.
- [50] Yehuda Koren, Robert Bell and Chris Volinsky. ‘Matrix factorization techniques for recommender systems’. In: *Computer* 42.8 (2009), pp. 30–37.
- [51] Pigi Kouki et al. ‘Generating and Understanding Personalized Explanations in Hybrid Recommender Systems’. In: *ACM Transactions on Interactive Intelligent Systems (TiiS)* 10.4 (2020), pp. 1–40.
- [52] Vaibhav Kumar et al. ‘User profiling based deep neural network for temporal news recommendation’. In: *2017 IEEE International Conference on Data Mining Workshops (ICDMW)*. IEEE. 2017, pp. 765–772.
- [53] Andrey Kutuzov and Maria Kunilovskaya. ‘Size vs. structure in training corpora for word embedding models: Araneum russicum maximum and russian national corpus’. In: *International Conference on Analysis of Images, Social Networks and Texts*. Springer. 2017, pp. 47–58.
- [54] Lei Li, Li Chen and Ruihai Dong. ‘CAESAR: context-aware explanation based on supervised attention for service recommendations’. In: *Journal of Intelligent Information Systems* (2020), pp. 1–24.
- [55] Jianxun Lian et al. ‘Towards Better Representation Learning for Personalized News Recommendation: a Multi-Channel Deep Fusion Approach.’ In: *IJCAI*. 2018, pp. 3805–3811.
- [56] Blerina Lika, Kostas Kolomvatsos and Stathes Hadjiefthymiades. ‘Facing the cold start problem in recommender systems’. In: *Expert Systems with Applications* 41.4 (2014), pp. 2065–2073.
- [57] Jionghao Lin and Yiren Liu. ‘A Neural Network Based Explainable Recommender System’. In: *arXiv preprint arXiv:1812.11740* (2018).
- [58] Yujie Lin et al. ‘Explainable outfit recommendation with joint outfit matching and comment generation’. In: *IEEE Transactions on Knowledge and Data Engineering* 32.8 (2019), pp. 1502–1516.
- [59] Peng Liu, Lemei Zhang and Jon Atle Gulla. ‘Dynamic attention-based explainable recommendation with textual and visual fusion’. In: *Information Processing & Management* 57.6 (2020), p. 102099.
- [60] Yinhan Liu et al. ‘Roberta: A robustly optimized bert pretraining approach’. In: *arXiv preprint arXiv:1907.11692* (2019).

- [61] Scott M Lundberg et al. ‘Explainable machine-learning predictions for the prevention of hypoxaemia during surgery’. In: *Nature Biomedical Engineering* 2.10 (2018), p. 749.
- [62] Julian McAuley and Jure Leskovec. ‘Hidden factors and hidden topics: understanding rating dimensions with review text’. In: *Proceedings of the 7th ACM conference on Recommender systems*. 2013, pp. 165–172.
- [63] Tomas Mikolov et al. ‘Efficient estimation of word representations in vector space’. In: *arXiv preprint arXiv:1301.3781* (2013).
- [64] Amy Mitchell, Galen Stocking and Katerina Eva Matsa. ‘Long-form reading shows signs of life in our mobile news world’. In: *Pew Research Center* 5 (2016).
- [65] Christoph Molnar. *Interpretable machine learning*. Lulu.com, 2020.
- [66] Grégoire Montavon et al. ‘Layer-wise relevance propagation: an overview’. In: *Explainable AI: interpreting, explaining and visualizing deep learning* (2019), pp. 193–209.
- [67] Douglas C Montgomery, Elizabeth A Peck and G Geoffrey Vining. *Introduction to linear regression analysis*. John Wiley & Sons, 2021.
- [68] Ramaravind K Mothilal, Amit Sharma and Chenhao Tan. ‘Explaining machine learning classifiers through diverse counterfactual explanations’. In: *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*. 2020, pp. 607–617.
- [69] Dat Quoc Nguyen, Thanh Vu and Anh Tuan Nguyen. ‘BERTweet: A pre-trained language model for English Tweets’. In: *arXiv preprint arXiv:2005.10200* (2020).
- [70] Michael Nielsen. *Neural Networks and Deep Learning*. <http://neuralnetworksanddeeplearning.com/>. 2019.
- [71] Ingrid Nunes and Dietmar Jannach. ‘A systematic review and taxonomy of explanations in decision support and recommender systems’. In: *User Modeling and User-Adapted Interaction* 27.3 (2017), pp. 393–444.
- [72] Shumpei Okura et al. ‘Embedding-based news recommendation for millions of users’. In: *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2017, pp. 1933–1942.
- [73] Lim CG Okura S Lee WJ and Choi HJ. ‘Embedding-based News Recommendation for Millions of Users’. In: (2017).
- [74] Alexis Papadimitriou, Panagiotis Symeonidis and Yannis Manolopoulos. ‘A generalized taxonomy of explanations styles for traditional and social recommender systems’. In: *Data Mining and Knowledge Discovery* 24.3 (2012), pp. 555–583.
- [75] Yoojin Park, Jinoh Oh and Hwanjo Yu. ‘RecTime: Real-Time recommender system for online broadcasting’. In: *Information Sciences* 409 (2017), pp. 1–16.
- [76] Georgina Peake and Jun Wang. ‘Explanation mining: Post hoc interpretability of latent factor models for recommendation systems’. In: *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2018, pp. 2060–2069.
- [77] 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)*. 2014, pp. 1532–1543.



- [78] Shaina Raza and Chen Ding. *News Recommender System: A review of recent progress, challenges, and opportunities*. 2021. arXiv: 2009.04964 [cs . IR] .
- [79] Nils Reimers and Iryna Gurevych. ‘Sentence-bert: Sentence embeddings using siamese bert-networks’. In: *arXiv preprint arXiv:1908.10084* (2019).
- [80] Marco Tulio Ribeiro, Sameer Singh and Carlos Guestrin. “ ‘ Why should i trust you?’ Explaining the predictions of any classifier’. In: *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. 2016, pp. 1135–1144.
- [81] Marco Tulio Ribeiro, Sameer Singh and Carlos Guestrin. ‘Model-agnostic interpretability of machine learning’. In: *arXiv preprint arXiv:1606.05386* (2016).
- [82] Francesco Ricci, Lior Rokach and Bracha Shapira. ‘Introduction to recommender systems handbook’. In: *Recommender systems handbook*. Springer, 2011, pp. 1–35.
- [83] Sungyong Seo et al. ‘Interpretable convolutional neural networks with dual local and global attention for review rating prediction’. In: *Proceedings of the eleventh ACM conference on recommender systems*. 2017, pp. 297–305.
- [84] Lloyd S Shapley. ‘A value for n-person games’. In: *Contributions to the Theory of Games* 2.28 (1953), pp. 307–317.
- [85] Jaspreet Singh and Avishek Anand. ‘Exs: Explainable search using local model agnostic interpretability’. In: *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*. 2019, pp. 770–773.
- [86] Yang Song, Ali Mamdouh Elkahky and Xiaodong He. ‘Multi-rate deep learning for temporal recommendation’. In: *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*. 2016, pp. 909–912.
- [87] Gabriel de Souza Pereira Moreira. ‘CHAMELEON: a deep learning meta-architecture for news recommender systems’. In: *Proceedings of the 12th ACM Conference on Recommender Systems*. 2018, pp. 578–583.
- [88] *Speech and Language Processing*. 2020. URL: <http://web.stanford.edu/%20CC%83jurafsky/slp3/>.
- [89] Natalie Jomini Stroud. ‘Polarization and partisan selective exposure’. In: *Journal of communication* 60.3 (2010), pp. 556–576.
- [90] Emily Sullivan et al. ‘Reading news with a purpose: Explaining user profiles for self-actualization’. In: *Adjunct Publication of the 27th Conference on User Modeling, Adaptation and Personalization*. 2019, pp. 241–245.
- [91] Panagiotis Symeonidis, Alexandros Nanopoulos and Yannis Manolopoulos. ‘Movi-Explain: a recommender system with explanations’. In: *Proceedings of the third ACM conference on Recommender systems*. 2009, pp. 317–320.
- [92] Pang-Ning Tan et al. *Introduction to Data Mining (2nd Edition)*. 2nd. Pearson, 2018. ISBN: 0133128903.
- [93] Nava Tintarev and Judith Masthoff. ‘Designing and evaluating explanations for recommender systems’. In: *Recommender systems handbook*. Springer, 2011, pp. 479–510.

- [94] Nava Tintarev and Judith Masthoff. ‘Effective explanations of recommendations: user-centered design’. In: *Proceedings of the 2007 ACM conference on Recommender systems*. 2007, pp. 153–156.
- [95] Laurens Van der Maaten and Geoffrey Hinton. ‘Visualizing data using t-SNE.’ In: *Journal of machine learning research* 9.11 (2008).
- [96] Sandra Wachter, Brent Mittelstadt and Chris Russell. ‘Counterfactual explanations without opening the black box: Automated decisions and the GDPR’. In: *Harv. JL & Tech.* 31 (2017), p. 841.
- [97] Hongwei Wang et al. ‘Ripplenet: Propagating user preferences on the knowledge graph for recommender systems’. In: *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. 2018, pp. 417–426.
- [98] Chuhan Wu et al. ‘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)*. 2019, pp. 6390–6395.
- [99] Chuhan Wu et al. ‘NPA: neural news recommendation with personalized attention’. In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2019, pp. 2576–2584.
- [100] Fangzhao Wu et al. ‘Mind: A large-scale dataset for news recommendation’. In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 2020, pp. 3597–3606.
- [101] Yao Wu and Martin Ester. ‘Flame: A probabilistic model combining aspect based opinion mining and collaborative filtering’. In: *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining*. 2015, pp. 199–208.
- [102] Shuai Zhang et al. ‘Deep learning based recommender system: A survey and new perspectives’. In: *ACM Computing Surveys (CSUR)* 52.1 (2019), pp. 1–38.
- [103] Yongfeng Zhang and Xu Chen. ‘Explainable recommendation: A survey and new perspectives’. In: *arXiv preprint arXiv:1804.11192* (2018).
- [104] Yongfeng Zhang et al. ‘Explicit factor models for explainable recommendation based on phrase-level sentiment analysis’. In: *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*. 2014, pp. 83–92.


# Appendix A

The online assessment tool "SurveyMonkey"<sup>1</sup> was used for collecting quantitative results through a user survey. The following pages depicts the survey in its entirety.

---

<sup>1</sup>[surveymonkey.com](https://www.surveymonkey.com)

## Appendix A - User Survey

 NTNU  
Kunnskap for en bedre verden

## Explainable Recommendations

Survey for evaluating explanations in a news recommender system.

**Dette er en spørreundersøkelse gjennomført i forbindelse med brukerevaluering av masteroppgave i datateknologi ved Institutt for Datateknologi og Informatikk ved NTNU.**

**Formålet med undersøkelsen er å evaluere ulike forklaringer i et anbefalingssystem for nyhetsartikler. Undersøkelsen samler ingen personopplysninger verken indirekte eller direkte, og alle individuelle besvarelser vil slettes innen 01.08.2021.**


**Introduksjon:**

Gjennom denne undersøkelsen vil du bli bedt om å vurdere ulike forklaringer i et anbefalingssystem for nyhetsartikler på engelsk.

Du vil først bli tildelt en rekke nyhetsartikler som inngår i din predefinerte leserhistorikk. Denne historikken er en reell leserhistorikk tatt fra en anonymisert bruker.

Videre vil du få presentert en anbefalt nyhetsartikkel, samt en tilhørende forklaring. Du skal vurdere forklaringene i henhold til oppgitte kriterier.

## Appendix A - User Survey



## Explainable Recommendations

### Lesehistorikk

**Vi antar at følgende nyhetsartikler inngår i din historikk. Gjør deg kort kjent med artiklene.**

#### 10 of the best fast-food burger chains across the US

Fast-food spots like Shake Shack and In-N-Out sell fresh burgers that keep customers coming back for more.

#### America shockingly owes over \$6 trillion to these countries

In 2019 America's debt surpassed \$22 trillion for the first time ever, as debt has climbed dramatically in the years following 2008's financial crisis. But who owns it? US debt can be bought by anyone through treasury securities (bills, notes or bonds), which represent an IOU from the government to the investor. The US then pays interest every year to service the debts. While much of America's debt is owed domestically, foreign investors hold more than \$6 trillion, both through central banks and private funds.

#### Reality television star Kevin O'Leary and his wife were sued Wednesday for wrongful deaths in a boat crash in Canada's backwoods.

Reality TV star Kevin O'Leary and his wife were sued Wednesday over an August boat crash in Canada's backwoods that killed two people and seriously injured three.

#### The News In Cartoons

News as seen through the eyes of the nation's editorial cartoonists.


#### Woman Spots Deadly Animal Hiding In Photo Of Her Kids

Her unsuspecting children weren't the only ones posing for the pictures.

#### Mitch McConnell snubbed by Elijah Cummings' pallbearer in handshake line at U.S. Capitol ceremony

A pallbearer appeared to refuse to shake Mitch McConnell's hand as Rep. Elijah Cummings was lying in state at the Capitol.

## Appendix A - User Survey

 NTNU  
Kunnskap for en bedre verden

**Explainable Recommendations**

**Eksempel:**  
På neste side vil du presenteres for en rekke anbefalte artikler med en tilhørende forklaring. Sammenlignet med typiske nettaviser så inneholder ikke artiklene et bilde, og du vil kun få presentert tittelen på artikkelen.

Article title

We recommend:


**"Video captures terrifying moment woman slips at Grand Canyon"**

Explanation:

**This is a viral article, similar to "Woman spots deadly animal hiding in photo of her kids"**

Explanation that aim to answer why this article was recommended

## Appendix A - User Survey

 NTNU  
Kunnskap for en bedre verden

## Explainable Recommendations

### Del 1 - Anbefalinger

1.

We recommend:

## "The Unraveling of Donald Trump"

Explanation:

## We think you would like this article

**Denne forklaringen..**

	Helt uenig	Uenig	Usikker	Enig	Helt enig
.. gjør at jeg vil lese artikkelen	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
.. øker min tiltro til anbefalingen	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
.. gjør at jeg forstår hva anbefalingen er basert på	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
.. ville latt meg gi konkret tilbakemelding på hvorvidt mine interesser er ivaretatt	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
.. ville gjort det enklere å velge anbefalte nyhetsartikler	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
.. hjelper meg å bestemme hvor godt jeg vil like artikkelen	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
.. gjør meg mer effektiv når jeg leser anbefalte nyhetsartikler	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

## Appendix A - User Survey

2.

We recommend:

**"The Unraveling of Donald Trump"**

Explanation:

**This is a newsopinion article.**

**Denne forklaringen..**

	Helt uenig	Uenig	Usikker	Enig	Helt enig
.. gjør at jeg vil lese artikkelen	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
.. øker min tiltro til anbefalingen	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
.. gjør at jeg forstår hva anbefalingen er basert på	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
.. ville latt meg gi konkret tilbakemelding på hvorvidt mine interesser er ivaretatt	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
.. ville gjort det enklere å velge anbefalte nyhetsartikler	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
.. hjelper meg å bestemme hvor godt jeg vil like artikkelen	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
.. gjør meg mer effektiv når jeg leser anbefalte nyhetsartikler	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>



## Appendix A - User Survey

3.

We recommend:

**"The Unraveling of Donald Trump"**

Explanation:

**This article is about newsopinion, which is amongst your interests.**

**Denne forklaringen..**

	Helt uenig	Uenig	Usikker	Enig	Helt enig
.. gjør at jeg vil lese artikkelen	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
.. øker min tiltro til anbefalingen	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
.. gjør at jeg forstår hva anbefalingen er basert på	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
.. ville latt meg gi konkret tilbakemelding på hvorvidt mine interesser er ivaretatt	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
.. ville gjort det enklere å velge anbefalte nyhetsartikler	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
.. hjelper meg å bestemme hvor godt jeg vil like artikkelen	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
.. gjør meg mer effektiv når jeg leser anbefalte nyhetsartikler	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

## Appendix A - User Survey

4.

We recommend:

**"The Unraveling of Donald Trump"**

Explanation:

**As the impeachment inquiry intensifies, some associates of the president predict that his already erratic behavior is going to get worse...**

**Denne forklaringen..**

	Helt uenig	Uenig	Usikker	Enig	Helt enig
.. gjør at jeg vil lese artikkelen	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
.. øker min tiltro til anbefalingen	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
.. gjør at jeg forstår hva anbefalingen er basert på	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
.. ville latt meg gi konkret tilbakemelding på hvorvidt mine interesser er ivaretatt	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
.. ville gjort det enklere å velge anbefalte nyhetsartikler	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
.. hjelper meg å bestemme hvor godt jeg vil like artikkelen	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
.. gjør meg mer effektiv når jeg leser anbefalte nyhetsartikler	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

## Appendix A - User Survey

5.

We recommend:

**"The Unraveling of Donald Trump"**

Explanation:

**This is a newsopinion article, similar to "The News In Cartoons", which you read previously.**

**Denne forklaringen..**

	Helt uenig	Uenig	Usikker	Enig	Helt enig
.. gjør at jeg vil lese artikkelen	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
.. øker min tiltro til anbefalingen	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
.. gjør at jeg forstår hva anbefalingen er basert på	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
.. ville latt meg gi konkret tilbakemelding på hvorvidt mine interesser er ivaretatt	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
.. ville gjort det enklere å velge anbefalte nyhetsartikler	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
.. hjelper meg å bestemme hvor godt jeg vil like artikkelen	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
.. gjør meg mer effektiv når jeg leser anbefalte nyhetsartikler	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

## Appendix A - User Survey

6.

We recommend:

**"The Unraveling of Donald Trump"**

Explanation:

**Because you read "News in Cartoons", which also mention Donald Trump.**

**Denne forklaringen..**

	Helt uenig	Uenig	Usikker	Enig	Helt enig
.. gjør at jeg vil lese artikkelen	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
.. øker min tiltro til anbefalingen	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
.. gjør at jeg forstår hva anbefalingen er basert på	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
.. ville latt meg gi konkret tilbakemelding på hvorvidt mine interesser er ivaretatt	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
.. ville gjort det enklere å velge anbefalte nyhetsartikler	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
.. hjelper meg å bestemme hvor godt jeg vil like artikkelen	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
.. gjør meg mer effektiv når jeg leser anbefalte nyhetsartikler	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

## Appendix A - User Survey

7.

We recommend:

**"The Unraveling of Donald Trump"**

Explanation:

**Your interest in "newsopinion" contributed 42% to this recommendation, while your interest in "politics" contributed 12%.**

**Denne forklaringen..**

	Helt uenig	Uenig	Usikker	Enig	Helt enig
.. gjør at jeg vil lese artikkelen	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
.. øker min tiltro til anbefalingen	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
.. gjør at jeg forstår hva anbefalingen er basert på	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
.. ville latt meg gi konkret tilbakemelding på hvorvidt mine interesser er ivaretatt	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
.. ville gjort det enklere å velge anbefalte nyhetsartikler	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
.. hjelper meg å bestemme hvor godt jeg vil like artikkelen	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
.. gjør meg mer effektiv når jeg leser anbefalte nyhetsartikler	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

8.

We recommend:

**"The Unraveling of Donald Trump"**

Explanation:

**The following features contributed to the recommendation:**

## Appendix A - User Survey

**Denne forklaringen..**

	Helt uenig	Uenig	Usikker	Enig	Helt enig
.. gjør at jeg vil lese artikkelen	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
.. øker min tiltro til anbefalingen	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
.. gjør at jeg forstår hva anbefalingen er basert på	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
.. ville latt meg gi konkret tilbakemelding på hvorvidt mine interesser er ivaretatt	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
.. ville gjort det enklere å velge anbefalte nyhetsartikler	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
.. hjelper meg å bestemme hvor godt jeg vil like artikkelen	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
.. gjør meg mer effektiv når jeg leser anbefalte nyhetsartikler	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>