Is attention explanation? A look at the case of sequential recommender systems

Thomas Baumhauer

Master of Science
Artificial Intelligence
School of Informatics
University of Edinburgh
2021

Abstract

Sequential recommender systems make predictions of what item a user is likely to interact with next, based on the sequence of the user's past interactions. Recently, Transformers [46] have been successfully employed to model interaction sequences [8]. Since it is well known that explaining recommendation to users increases satisfaction with recommender systems [57] the question arises of how to explain predictions made by Transformer-based sequential recommender systems. Because Transformers employ attention mechanisms which are often claimed to provide insights into the workings of models (e.g. [31]) it is natural to ask whether attention could be used to explain these recommender systems. Recently, [25] claimed that "attention is not explanation", which initiated a discussion among researchers on the subject, mostly in the context of natural language processing. This dissertation contributes to this discussion for the domain of recommender systems. Our main result is that by employing the attention rollout algorithm [1], a tool originally developed for Transformers in computer vision, we are able to construct an attention-based importance ranking which strongly correlates with a post-hoc importance ranking based on gradients. This challenges one of the central arguments of [25] who were not able to related attention to established post-hoc methods. Thus, our work makes a case for attention as explanation for Transformer-based sequential recommender systems.

Acknowledgements

I am thankful to my supervisors Andrei Apostoae and David Wardrope for sharing their invaluable insights and suggestions during the undertaking of my dissertation project. Their high engagement and prompt feedback was highly appreciated and substantially improved the quality of this work.

I am thankful to the Esk House crew, Carel, Ondre, and Zixu, as well as the awesome people of the Hare and Hounds running club for making the winter semester as enjoyable as was possible given the circumstances. Likewise, I am thankful to my family for supporting me while I was studying from home during the summer semester.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(*Thomas Baumhauer*)

Table of Contents

Chapter 1

Introduction

1.1 Background & Motivation

1.1.1 Recommender systems

Recommender systems are software tools that suggest items to users, based on recorded past interactions between items and users such as ratings, clicks, or purchases [43]. Typical applications include e-commerce platforms suggesting products for customers to purchase, or streaming services suggesting movies for subscribers to watch. Originally, recommender systems were often relatively straightforward matrix factorisation models [28, 41], computing latent representations of users and items and suggesting items based on the dot-product similarity between those representations. Nowadays, recommender systems are often complex AI models employing deep neural network architectures, e.g. [20, 10, 23, 22, 55].

Sequential recommender systems model which item a user is likely to interact with next, given a sequence of items a user interacted with previously [26, 53]. Recently, it has been proposed to build sequential recommender systems based on Transformers [59], a neural network architecture originally developed for natural language modeling. Transformer-based sequential recommender systems have been shown to outperform other sequence modeling tools such as Markov chains and recurrent neural networks [24] according to performance metrics such as recall [26]. Furthermore, they have been successfully employed in practice, e.g. to boost metrics such as the click-through rate of suggested items of a large e-commerce platform [8].

1.1.2 Explainable AI

Providing users with *explanations* for why a certain item was suggested to them has long been known to increase user satisfaction with recommender systems [57]. Considering the promising results for Transformer-based sequential recommender systems it is therefore natural to ask: how can we explain predictions made by these models?

It has been observed that it is in fact often difficult to understand how complex AI systems (such as recommender systems) arrive at their predictions, i.e. that they act as *black-boxes* [42]. Lack of explainability has been recognised as a serious issue, not only in the field of recommender systems [63] but also many other areas of AI, because AI systems are increasingly also deployed in high-stakes applications such as medicine and criminal justice [44]. This gave rise to the wide field of *explainable artificial intelligence*, the study of how to best construct explanations for the predictions made by AI systems [37]. In this dissertation, we follow the taxonomy and terminomolgy of [32], grouping explainability into two broad categories:

- *Intrinsically* explainable models for which their inner workings are (at least partially) accessible to human cognition (e.g. sufficiently small decision trees or low dimensional linear models). Such models are called *transparent* and are considered the opposite of a black-box model.

- *Post-hoc* methods for explainability which seek to extract further information beyond the raw prediction scores from a black-box model. Typical examples of post-hoc explanations are importance ranking of input features, e.g. calculated from the features' gradients.

1.1.3 Attention as explanation

The Transformer architecture employed by the sequential recommender systems we consider in this thesis is characterised by its use of *attention mechanisms* [3]. Roughly, attention mechanisms are a tool to aggregate sequences of vectors in a convex sum, with the summation weights (or "attention weights") dynamically calculated depending on the context. Semantically, they can thus be considered a tool for a model to learn which elements in a sequence to "pay attention to". This suggests that attention mechanisms could provide a window to discover and explain the inner workings of models. This idea is popular and is often taken as self-evident, e.g. [31] claims: "Attention provides an important way to explain the workings of neural models". Likewise, the survey paper [14] (despite being aware of the works subsequently discussed here indicating to the contrary!) states: "The attention mechanism can also be used as a tool for interpreting the behavior of neural architectures, which are notoriously difficult to understand". Indeed, there exist a large number of works in the literature based on similar claims that attention is to be equated with explanation, e.g. [62, 9, 34, 61, 29].

Using attention as explanation is certainly appealing. First, if attention could indeed provide explanations these explanations would be model-intrinsic and thus faithful to the models workings by construction, unlike explanations constructed post-hoc where one has to argue this point. Second, presenting attention weights as explanation for predictions is cheap since they need to be computed during the forward pass from which the predictions result anyway.

The idea of attention as explanation was challenged in the work "Attention is not Explanation" [25], where the authors find that 1) attention weights are frequently uncorrelated with established methods of ranking feature importance and 2) and attention weights can be adversarially manipulated. The latter is an important concern in the field of natural language processing, e.g. when using attention to gender pronouns as a tool to reveal potential model bias [12]. The work of [25] initiated a discussion among researchers, primarily focusing on their second point. In [39] the authors introduce an method for adversarially decreasing attention weights assigned to a predefined set of impermissible token, while in [60] the authors argue that "attention is not not explanation" and claim [25] allowed themselves too many degrees of freedom when constructing their adversarial attention weights.

1.2 Contribution

This dissertation contributes to the discussion whether attention is suitable as explanation in the context of sequential recommender systems. We focus on the first point of [25], the relationship of attention to other feature importance ranking methods, as we consider the issue of basing decision on impermissible features less relevant for the domain of recommender systems than it undoubtedly is for natural language processing.

Similarly to [25] we compare established post-hoc item importance ranking methods from the explainable AI literature to rankings obtained from attention weights. We conduct extensive experiments with two different state-of-the art sequential recommender models on a variety of datasets. Our main result is that the approaches found in works like [25, 60, 39, 46] to construct explanations from attention may be too naive. These approaches typically consider attention over intermediate representations containing information from multiple items (as pointed out by [39]), and do not take into account the model's specific architecture. Instead, we use the *attention rollout* algorithm [1] to construct item importance ranking from the attention weights, which combines attention scores from all intermediate layers while also taking into account model specifics such as skip-connections. In our experiments the attention rollout importance ranking achieves at least moderate and in almost all cases very strong rank correlation with a standard gradient-based importance ranking. For the case of sequential recommender systems our experiments thus contradict the argument of [25] that "attention is not explanation".

Additionally, we strengthen our case for attention as explanation for the domain of recommender systems by an experiment similar one performed by [46]. We gradually delete the top ranked items according to different importance metrics and observe changes in the model's predictions to measure "explanatory power" of these rankings. In this experiment attention performs comparable to a gradient-based ranking for our sequential recommender models, unlike in the experiments of [46] where attention was found to have less explanatory power than the gradients for text classification models.

As a methodological contribution, we propose a novel metric that can be used to measure if two rankings agree more on their top ranks. We use this metric to find evidence for a conjecture of [25] that while different importance rankings may agree on which items are the most important, a (potentially large) number of irrelevant features may depress traditional rank correlation metrics.

1.3 Outline

The remainder of this document is structured as follows. In Section 2 we describe the works mentioned in Section 1.1.3 which are most relevant to this dissertation in additional detail. In Section 3.1 we describe the datasets and in Section 3.2 the sequential recommender models we use in our experiments. In Section 3.3 we describe the item importance rankings employed. In Section 3.4 we introduce the Spearman rank correlation, as well as our novel metric to identify "top-heavy" agreement between rankings. Section 4 reports our experimental measurements. Finally, in Section 5 we interpret our results and argue that they are evidence for the conclusions outlined above. Furthermore, Section 5 also discusses the limitation of our work.

Chapter 2

Related work

2.1 Natural language processing

In Section 1.1.3 we briefly described several recent papers that investigate whether attention is suitable as explanation for the domain of natural language processing. In this section we add additional detail to their description.

In [25] the authors compare the importance ranking obtained from an attention mechanisms built into BiLSTM models trained on natural language processing tasks with gradient-based and leave-one-out importance rankings. They find that the rank correlations between these rankings are usually weak, which they conclude “ought to give pause to practitioners” looking to employ attention as explanation. In a second experiment they take and hold fixed trained models and use gradient ascend post-hoc to find different attention weights that produce the same predictions as the original weights. This approach has been criticised by [39], who argue that since these alternate attention weights are chosen from infinite set (up to numerical precision) the existence of these weights is not surprising, and since they were not produced by the model they would not be considered as explanations anyway.

Consequently, [39] devise an experiment in which they train adversarial models where attention to a selected set of impermissible tokens (such as gender pronouns) is penalised in the loss function. They show that these adversarial models come close to non-adversarial models in prediction performance, yet are able to deceive users about their reliance on gender pronouns (with the reliance shown by comparing the prediction accuracy to models trained on the same datasets with gender pronouns anonymised). Similarly, in [60] the authors train adversarial models to produce maximally different attention weights to the attention weights of non-adversarial models (according to the

Jensen–Shannon divergence) while producing the same predictions.

In [46] the authors conduct an experiment in which they delete the items with the highest attention values from the input sequence and observe the effect on the model's predictions. They find that the number of items that need to be deleted for the prediction to change is often large, and in particular larger than when deleting items according to a gradient-based ranking. While their verdict on attention is not as definite as that of [25], they conclude that "while attention noisily predicts input components' overall importance to a model, it is by no means a fail-safe indicator". We conduct a similar deletion experiment in Section 4.2.1.

2.2 Computer vision

There exists a small, very recent line of work on attention as explanation for Transformers for vision tasks.

In [1] the authors introduce the *attention rollout* algorithm to trace attention throughout the transformer architecture. They conduct experiments similar to those in this dissertation, comparing the attention rollout scores to leave-one-out estimates, observing weak to moderate rank correlations. Interestingly, they also state "it is wrong to equate attention with explanation" citing [25], seemingly unaware that they discovered a tool to challenge such claims as our experiments will show (at least for the domain of recommender systems). In this sense, the contribution of this dissertation could be framed as combining the intuition of [25] to relate attention to gradient-based importance metrics with the tools of [1].

In [6, 7] the authors propose a gradient weighted version of the attention rollout, showing improvements according to vision specific metrics over the vanilla attention rollout algorithm. We do not make use of this algorithm in this dissertation, since our preliminary results were not promising for the recommender systems domain.

Chapter 3

Method

3.1 Datasets

We employ the following real-world datasets of varying domains in our experiments, which are commonly used in the sequential recommender systems literature [26, 53, 8].

MovieLens [15]: The MovieLens datasets are some of the most popular benchmark datasets for recommender systems. They feature movie ratings collected online by the MovieLens project. We use the MovieLens 1 million (**ML-1M**) version of these datasets.

Amazon [35, 19]: These datasets contain product ratings organised by product categories collected from the e-commerce platform *amazon.com*. We use the datasets corresponding to the categories beauty products (**Beauty**) and video games (**Games**).

Steam [26]: This dataset contains ratings of video games collected from the online video games distribution platform *steam.com*.

Following [18, 17, 26, 53] these datasets are preprocessed as follows[1]: First, we convert all explicit rating feedback to implicit feedback by considering every rating to be an interaction. We then remove all users with less than 5 interactions from the dataset. For each user, we order all interactions by their timestamp, resulting in a sequence of items the user interacted with. For each user, we reserve the most recent item of testing, and the second most recent item for validation. In this work, we report all measurements for the validation set, since we do not use it to tune our models but instead copy settings from the literature. Beyond timestamps, we make no use of any other meta data (user demographics, item properties etc.) present in either of the

[1] The authors of [26] have released preprocessed versions of these datasets available at `https://github.com/kang205/SASRec/tree/master/data`.

Dataset	Users	Items	Actions	Median actions per item	Median actions per user	Density
ML-1M	6,040	3,416	999,611	146	96	4.845%
Steam	334,730	13,047	3,686,172	33	7	0.084%
Beauty	52,024	57,289	394,908	3	6	0.013%
Games	31,013	23,715	287,107	5	6	0.039%

Table 3.1: Statistics of the datasets used in this dissertation (after preprocessing according to [26]).

datasets, with the exception of Section 4.2.3 where we briefly investigate how genres of movies in the ML-1M dataset interact with the attention mechanism of the SASRec model.

The resulting statistics of the datasets can be found in Table 3.1. We observe that the datasets do not only vary in their domain, but also in their size and sparsity: ML-1M is the most *dense* (meaning that it has the largest number of user-item interaction relative to all possible user-item pairs.). Amazon Beauty and Games are very *sparse* (the opposite of dense). The Steam dataset sits between the ML-1M and Amazon datasets in sparsity and in particular has a much larger number interactions per item than the Amazon datasets. By choosing datasets of varying characteristics for our experiments we hope to increase the likelihood that any trend we observe across all of them is indeed a general trend (and not the result of specific properties of those datasets).

3.2 Models

Transformers [59] are a neural network architecture for sequence modeling, originally developed for natural language processing tasks. In this section we describe two different sequential recommender systems based on transformers: the unidirectional *SASRec* by [26] in Section 3.2.2 and the bidirectional *Bert4Rec* by [53] in Section 3.2.3.

3.2.1 Transformer building blocks

We begin by describing the building blocks of the transformer architecture, with some minor modifications of the original architecture, as we use them in in our models. In particular, the original transformer architecture described in [59] is more general.

3.2.1.1 Multi-head attention

The key characteristic of transformers is its use of attention mechanisms throughout the model. Transformers use *scaled dot-product attention*: Given a sequence of n-many d-dimensional queries $\mathbf{Q} \in \mathbb{R}^{n\times d}$, a sequence of n-many d-dimensional keys $\mathbf{K} \in \mathbb{R}^{n\times d}$, and a sequence of values $\mathbf{V} \in \mathbb{R}^{n\times d}$ (with these sequences being linear projections of some input sequence $\mathbf{S} \in \mathbb{R}^{n\times d}$, see below) scaled dot-product attention is defined as

$$\text{Attention}(\mathbf{Q},\mathbf{K},\mathbf{V},\mathbf{M}) = \text{softmax}(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d}} \odot \mathbf{M}) \cdot \mathbf{V}$$

where $\mathbf{M} \in \{0,1\}^{n\times n}$ is a binary mask encoding which sequence elements may interact with each other. E.g. in the SASRec model we will use a lower triangle matrix $\mathbf{M}$ to force items to only interact with other items that ocured previously in the input sequence. As $\mathbf{M}$ is typically a fixed model parameter we are mostly going to omit it in our notation. A single attention layer in a transformer consists of h attention heads. The i-th head computes

$$\text{head}_i = \text{Attention}(\mathbf{Q}\mathbf{W}_i^Q, \mathbf{K}\mathbf{W}_i^K, \mathbf{V}\mathbf{W}_i^V) \quad (i < h)$$

with projection matrices $\mathbf{W}_i^Q, \mathbf{W}_i^K, \mathbf{W}_i^V \in \mathbb{R}^{d\times d}$. These are then combined to

$$\text{Attention\_layer}(\mathbf{Q},\mathbf{K},\mathbf{V}) = \text{concatenate}_{i<h}(\text{head}_i)\mathbf{W}^O$$

with $\mathbf{W}^O \in \mathbb{R}^{d\times d}$ another projection matrix.

3.2.1.2 Stacking attention layers

The attention layers described above are then interleaved with simple point-wise feed-forward networks, which for an input sequence $\mathbf{S} \in \mathbb{R}^{n\times d}$ are defined as

$$\text{FFN}(\mathbf{S}) = \text{LeakyReLU}(\mathbf{S}\mathbf{W}_1 + \mathbf{b}_1)\mathbf{W}_2 + \mathbf{b}_2.$$

with $\mathbf{W}_1, \mathbf{W}_2 \in \mathbb{R}^{d\times d}$, $\mathbf{b}_1^\top, \mathbf{b}_2^\top \in \mathbb{R}^d$. Note that the original transformer architecture uses standard ReLU activations.

Attention and feed-forward layers are then combined into self-attention blocks, employing *residual connections* [16], *dropout* [52], and *layer normalisation* [2]:

$$g(\mathbf{S}) = \text{Dropout}(\text{Attention\_layer}(\mathbf{S},\mathbf{S},\mathbf{S})) + \mathbf{S}$$

$$\text{Block}(\mathbf{S}) = \text{LayerNorm}(\text{Dropout}(\text{FFN}(\text{LayerNorm}(g(\mathbf{S})))) + g(\mathbf{S})).$$

3.2.1.3 Embedding layers

The inputs to our transformer models are sequences of discrete item ids. Consequently, we employ learned *item embeddings* [36] $i \mapsto \mathbf{e}_i \in \mathbb{R}^d$ to construct our input sequence $\mathbf{S}^{(0)}$. Similarly, to keep track of the position of the j-th item in an input sequence of n items, we use positional embeddings $j \mapsto \mathbf{p}_j \in \mathbb{R}^d$. The dense embedding $\mathbf{S}^{(0)}$ of an input sequence of item ids $\langle i_1, i_2, \ldots, i_n \rangle$ is thus defined as

$$\mathbf{S}^{(0)} = \begin{pmatrix} \mathbf{e}_{i_1}^\top + \mathbf{p}_1^\top \\ \mathbf{e}_{i_2}^\top + \mathbf{p}_2^\top \\ \cdots \\ \mathbf{e}_{i_n}^\top + \mathbf{p}_n^\top \end{pmatrix}.$$

Note that in the original transformer architecture in [59] the positional embeddings are not learned, but statically defined based on trigonometric functions. However, the sequential recommender literature finds that learned positional embeddings work better for recommender systems [26, 53, 8].

3.2.2 SASRec

The SASRec model [26] stacks b attention blocks $\text{Block}_1, \ldots \text{Block}_b$. Given an input sequence of item ids $\langle i_1, i_2, \ldots, i_n \rangle$ it computes

$$\mathbf{S}^{(k)} = \text{Block}_k(\mathbf{S}^{(k-1)})$$

for $1 \leq k \leq b$ with $\mathbf{S}^{(0)}$ as defined in section 3.2.1.3. We denote the n rows of $\mathbf{S}^{(b)}$ by $\mathbf{h}_1^\top, \ldots \mathbf{h}_n^\top$ and interpret them as item embeddings. Consequently, the SASRec model predicts

$$\hat{i} = \text{argmax}_i \; \mathbf{h}_n^\top \mathbf{e}_i$$

as the continuation of the input sequence. We denote

$$\mathbf{H} = \langle \mathbf{h}_1^\top, \ldots \mathbf{h}_n^\top \rangle = \text{Enc}(\langle i_1, i_2, \ldots, i_n \rangle).$$

Loss. For training, the loss for an input sequence $\langle i_1, i_2, \ldots, i_n \rangle$ and a training label i^* (the item id following the input sequence in the training data) is computed as follows. First, sample n negative item ids $\langle i_2^-, i_3^-, \ldots, i_{n+1}^- \rangle$ uniformly random from the items the user who generated the input sequence has not interacted with. Denote $i_{n+1} = i^*$. The loss L is then computed as

$$L(\langle i_0, i_1, \ldots, i_n \rangle, i^*) = \sum_{j=1,2,\ldots,n} -\log(\sigma(\mathbf{h}_j^\top \mathbf{e}_{i_{j+1}})) - \log(1 - \sigma(\mathbf{h}_j^\top \mathbf{e}_{i_{j+1}^-}))$$

i.e. the model is trained to compute a left shift in the item sequence, with the last item consequently being the predicted continuation. To prevent the model from peeking to the right (which would make its task trivial for all but the last item) we use a lower triangle matrix $\mathbf{M}$ in the attention maps throughout all b blocks.

The authors of [26] find that their SASRec model “outperforms various state-of-the-art sequential models (including Markov chain/CNN/RNN-based approaches) on both sparse and dense datasets”.

3.2.3 BERT4Rec

Like the SASRec model, the BERT4Rec model [53] consists of stacked attention blocks. However, BERT4Rec is bidirectional, using $\mathbf{M} = \{1\}^{n \times n}$. During training, given and input sequence $\langle i_1, i_2, \ldots, i_n \rangle$, each item is “masked” with probability γ. Masked items are replaced by a special item `[CLS]` and the model is trained to reconstruct the embeddings of the masked items in the corresponding $\mathbf{h}_j$ of the output sequence. This is known as a *cloze task* [56] (employed also by the well-known language representation model BERT [13], hence the name BERT4Rec). To make predictions, we simply take an input sequence and append the `[CLS]` item to its right.

Loss. Let $C \subseteq \{1, 2, \ldots n\}$ denote the set of masked indices. During training the loss is defined as

$$L(\langle i_0, i_1, \ldots, i_n \rangle, C) = \sum_{j \in C} \text{cross\_entropy}(i_j, \text{softmax}(\mathbf{h}_j^\top \mathbf{E}))$$

where $\mathbf{E} = (\mathbf{e}_1 | \mathbf{e}_2 | \ldots | \mathbf{e}_{\#\text{items}})^\top$ is a matrix of all item embeddings. The authors of [53] do not study the effect of the cross-entropy loss across all items versus computing the loss only from the positive and a single sampled negative item as for the SASRec model described above. However, anecdotally we find this choice does indeed help the performance of the BERT4Rec model (even though it of course comes at considerably increased computational costs during training).

In their experiments [53] find that BERT4Rec outperforms SASRec consistently. As we are going to see, this does not appear to be a general trend in our implementations. However, since we do not focus on finding the optimal configurations for SASRec and BERT4Rec for performance but rather run them in varying configurations to explore the configurations’ effect on importance rankings for explainability, our findings do not contradict this result.

Model	Dataset	$b=1$ $h=1$	$b=2$ $h=1$	$b=3$ $h=1$	$b=2$ $h=2$
SASRec	ML-1M	0.81	0.82	0.82	0.82
	Steam	0.70	0.70	0.69	0.70
	Beauty	0.44	0.43	0.44	0.44
	Games	0.61	0.60	0.60	0.61
BERT4Rec	ML-1M	0.77	0.79	0.80	0.79
	Steam	0.68	0.69	0.70	0.69
	Beauty	0.42	0.44	0.45	0.44
	Games	0.62	0.63	0.62	0.62

Table 3.2: Recall@10% for b attention blocks and h heads per attention block.

3.2.4 Training setup

We limit the input sequence length to $n = 50$ items for the ML-1M dataset and $n = 25$ for the remaining datasets. While [26, 53] use 200 and 50 items respectively, they also find that such long input sequences only increase model performance marginally. Therefore, we choose to work with shorter sequences which are more appropriate to our computational budget. If the original item sequence is longer than the maximum length, we sample a random subsections during training. If it is shorter, we pad it from the left to the maximum length n and set the corresponding columns in the mask $\mathbf{M}$ to 0. Following [26] we use a batch size of 128 and a learning rate of 10^{-3} for the Adam optimiser [27], a dropout rate of 0.2, and latent dimension $d = 64$. Following [53], for the BERT4Rec models we use $\gamma = 0.2$ for the ML-1M datset, $\gamma = 0.4$ for the Steam dataset, and $\gamma = 0.6$ for the remaining datasets.

During training, we sample (in expectation) from each user's item sequence 50 times for the ML-1M dataset, 10 times for the Beauty and Games and 2 times for the Steam dataset. We arbitrarily divide the training into 10 "epochs" for the investigation of training dynamics in Section 4.1.4, resulting in 236 batches per epoch for ML-1M, 406 for Beauty, 242 for Games, and 523 for Steam. Most models converge before 10 epochs are up, however as we are going to see Section 4.1.4 the dynamics of explainability methods may take longer. However, the regularisation strength of the dropout layers used appears to be sufficient for overfitting not to become an issue.

Table 3.2 reports achieved recall@10% for the model types, datasets and configurations we use. The recall@10% metric is commonly employed in the sequential recommender literature [24, 26, 53] and is computed by sampling 100 random items, ranking them and the ground-truth item by their likelihood as a continuation according to the model and taking the fraction of times the ground-truth item is contained in the top 10 highest ranked items.[2] The recall@10% scores are consistently slightly lower than those achieved by [26, 53] which is reasonable since we do not exactly replicate their optimal configurations as described above. These results are also in line with the observations of [8, 53, 26] that for sequential recommender systems comparatively few attention blocks, as well as attention heads per block are needed (suggestions in the literature are typically 1 or 2 blocks as well as 1 or 2 heads, while the original transformer model uses $b = 6$ blocks and $h = 8$ heads [59]).

3.3 Importance rankings for explainability

In this section we introduce methods to rank item importance for trained models. First, we introduce the attention rollout (AR) algorithm [1] used to combine the attention scores of multiple attention blocks into a single importance ranking. We then describe two established importance rankings from the explainability literature: Shapley values [48, 33] in Section 3.3.3 and a gradient based measure [30, 38] in Section 3.3.4.

3.3.1 Attention

Each head in each attention block computes an attention map

$$\text{softmax}\left(\frac{\mathbf{QK}^\top}{\sqrt{d}} \odot \mathbf{M}\right).$$

We denote the map corresponding to the i-th head of the j-th attention block by $\mathbf{A}_i^{(j)}$. In case the block only has a single head we omit the index i. For a model consisting of b blocks, we consider the importance rankings

$$\alpha_{\text{first}} = \frac{1}{h}\Big(\sum_{1\leq i\leq h} (A_i^{(1)})_{n,\cdot}\Big),$$

$$\alpha_{\text{last}} = \frac{1}{h}\Big(\sum_{1\leq i\leq h} (A_i^{(b)})_{n,\cdot}\Big).$$

These are the means of the last rows of the attention maps of the first and last attention blocks, respectively. These rows corresponds to the last item in the transformer

[2] Confusingly recall@10% is sometimes called recall@10 in the sequential recommender literature.

sequence, which for either model type corresponds to the prediction. However, since for the BERT4Rec model the last element of the input sequence is the `[CLS]` token, when referring to attention scores for the BERT4Rec model we abuse notation and take α_{first} to denote only the first $n-1$ elements of α_{first} in the sense defined above, and likewise for α_{last}, as these are the attention scores corresponding to the input sequence.

Sofar, previous work investigating attention as explanation mostly focused on attention over intermediate representations [25, 60, 46, 40] conceptually most closely related to α_{last} which is why we include this method in our experiments.

3.3.2 Attention rollout

A more sophisticated way to construct importance rankings from raw attention scores is the *attention rollout* (AR) recently proposed by [1]. The idea is to "trace" the attention between items through all attention blocks. E.g. in a transformer using $b = 2$ blocks, to compute how much attention the first element of the output sequence $\mathbf{h}_1$ pays to the first element of the input sequence we first look at the row $\mathbf{A}^{(2)}_{1,\cdot}$ to see how much attention is payed to each intermediate output of the first block. Then, we look for much how attention each intermediate representation paid to the first item of the input sequence, which we can read out by looking at the column $\mathbf{A}^{(1)}_{\cdot,1}$. For now ignoring the skip connections, the traced total attention paid to the first item is thus $\mathbf{A}^{(2)}_{1,\cdot}\mathbf{A}^{(1)}_{\cdot,1}$.

Doing this calculation for all item pairs simultaneously, repeating the calculation for each attention block, averaging across heads, and taking into account the skip connections (which imply that at each position least half the attention in each block is always paid to the position itself) the attention rollout $\tilde{\mathbf{A}}$ is defined as

$$\tilde{\mathbf{A}} = \prod_{b \geq i \geq 1} \left(\frac{1}{2h}\Big(\sum_{1 \leq j \leq h} \mathbf{A}^{(i)}_j\Big) + \frac{1}{2}\mathbf{I}_n\right).$$

As we did for the raw attention scores, we define $\alpha = \tilde{\mathbf{A}}_{n,\cdot}$ to be the last row of the attention rollout corresponding to the predicted item. Again, as in Section 3.3.1, when referring to the scores for BERT4Rec we discard the attention rollout score associated with the `[CLS]` token and take α to only denote the first $n-1$ attention rollout scores. Note, that for $b = 1$ attention blocks for the BERT4Rec model this implies $\alpha = \alpha_{\text{first}} = \alpha_{\text{last}}$.

In [6, 7] the authors investigate a gradient weighted version of the attention rollout. They rely primarily on vision task specific metrics such as weakly supervised segmentation performance to demonstrate the improvements achieved by their method. Since our initial results using gradient weighting in the attention rollout were not promising, we chose to stick to the vanilla version of the attention rollout described in this section. Furthermore, as an alternative to the attention rollout algorithm [1] also propose *attention flow*, which tracks attention based on a graph theoretical maximum flow formulation. As point [6] point out the attention flow algorithm is too computationally expensive in practice, hence we again opt for the attention rollout algorithm instead.

3.3.3 Shapley values

Shapley values are a decompositional importance ranking with a theoretical foundation in game theory [48]. For our transformer models they are defined as follows. Let $\langle i_1, i_2, \ldots, i_n \rangle$ be an input sequence of item ids and let i^* the items for which we want to explain the model's prediction of its likelihood as a continuation. For $S \subseteq \{1, 2, \ldots n\} = N$ let $\mathbf{M}(S) = \mathbf{M} \cdot \operatorname{diag}(\langle \mathbb{1}_{[j \notin S]} : j = 1, 2, \ldots, n \rangle)$, where $\mathbf{M}$ is the default mask used by the model (lower triangle for SASRec and ones for BERT4Rec) and $\mathbf{M}(S)$ is the mask $\mathbf{M}$ with the columns not in S zeroed out, i.e. only items with indices in S are considered. Let $f(S) = \mathbf{h}_n(S)^\top \mathbf{e}_{i^*}$ where $\mathbf{h}_n(S)$ is the predicted item embedding calculated with mask $\mathbf{M}(S)$. The Shapley value of the j-th item is defined as

$$\varphi_j = \sum_{S \subseteq N \setminus \{j\}} \frac{|S|!(n - |S| - 1)!}{n!} (f(S \cup \{j\}) - f(S))$$

and measures the contribution of the j-th item to the model's prediction. The Shapley values can be computed as the solution to a weighted linear regression [5] (as pointed out in the context of explainable AI by [33]). We employ a Monte Carlo version of this computation listed in Algorithm 1. The reason we only compute a Monte Carlo estimate of the Shapley values is that computing them exactly, i.e. enumerate all $a = 2^n$ possible labels is infeasible for all but the smallest n. In our experiments we use $a = 1024$ (= 8 batches of size 128).

We choose to consider Shapley values as they are a very natural importance ranking for sequential recommender systems, since input sequences can easily be decomposed into present and absent items. However, Shapley values are a *signed* importance ranking in the sense that they measure positive and negative contributions of items, while attention is unsigned so that we can only hope for attention to measure the magnitute of the contribution of items to predictions. We therefore use the absolute Shapley values $|\varphi| = \langle |\varphi_1|, |\varphi_2|, \ldots, |\varphi_n| \rangle$ in our experiments.

Algorithm 1 Monte Carlo estimation of Shapley values for Transformers

```
 Input Item sequence $\langle i_1, i_2, \dots i_n \rangle$, target item $i^*$, number of samples $a$.
 Output Estimated Shapley values $\varphi = \langle \varphi_1, \varphi_2, \dots \phi_n \rangle$.
Initialise empty arrays labels, values, label_weights.
for $k = 1, 2, \dots, a$ do
    $\mathbf{m} \leftarrow$ sample_uniform_random($\{0,1\}^n$)
    $S \leftarrow \{j \in N : m_j = 1\}$
    $\langle \mathbf{h}_1^\top, \mathbf{h}_2^\top, \dots, \mathbf{h}_n^\top \rangle \leftarrow \mathrm{Enc}(\langle i_1, i_2, \dots i_n \rangle, \mathbf{M}(S))$
    $labels[k] = \mathbf{m}$
    $values[k] = \mathbf{h}_n(S)^\top \mathbf{e}_{i^*}$
    $label\_weights[k] = \frac{(n-1)}{(n \text{ choose } \|\mathbf{m}\|_1)\|\mathbf{m}\|_1(n-\|\mathbf{m}\|_1)}$
end for
weights, bias $\leftarrow$ weighted_linear_regression(labels, values, label_weights)
$\varphi \leftarrow$weights
return $\varphi$
```

3.3.4 Gradients

A popular method in explainable AI to construct importance rankings for differentiable models is using gradients, e.g. [50, 51, 54]. The general underlying intuition is that if predictions are sensitive to certain input features, those features are important for the model's prediction.

For our transformer models we define a gradient-based importance ranking, similar to the one used in [25], as follows. Let $\langle i_1, i_2, \dots, i_n \rangle$ be the input sequence of item ids and let i^* the item for which we want to explain the model's prediction of its likelihood as a continuation. Again following the notation of Section 3.2.2 we define the importance of the j-th item in the input sequence as

$$\nabla_j = \left\| \frac{\partial \mathbf{h}_n^\top \mathbf{e}_{i^*}}{\partial \mathbf{e}_{i_j}} \right\|_2$$

i.e. the sensitivity of the alignment of the predicted embedding $\mathbf{h}_n$ with the embedding $\mathbf{e}_{i^*}$ of the item i^* with respect to the embedding $\mathbf{e}_{i_j}$ of the j-th item.

3.4 Rank-based similarity measures

In this section we introduce metrics to compare importance rankings and discuss why we consider them appropriate tools for our purposes.

3.4.1 Spearman rank correlation

Given a sequence of pairs of observations X_i, Y_i from two random variables X, Y, denote the ranks within the observation from either variable by r_{X_i}, r_{Y_i}. The Spearman rank correlation coeffcient is then defined as the correlation of the rank variables

$$\rho_{\text{Spearman}}(X,Y) = \frac{\text{cov}(r_X, r_Y)}{\sigma_{r_X}\sigma_{r_Y}}$$

where $\text{cov}(r_X, r_Y)$ denotes the covariance and $\sigma_{r_X}\sigma_{r_Y}$ denote the standard deviations of the rank variables. Unlike the Pearson correlation coefficient

$$\rho_{\text{Pearson}}(X,Y) = \frac{\text{cov}(X,Y)}{\sigma_X\sigma_Y}$$

which measures linear dependence between X, Y, the Spearman rank correlation measures arbitrary monotonic dependence. It takes values in $[-1, 1]$ with the extremes 1 denoting that Y is a monotonic increasing function of X and -1 denoting that Y is a monotonic decreasing function of X. Informally, we refer to the rank correlation as weak, moderate, or strong, if its magnitude is roughly in $[0, .3), [.3, .7), [.7, 1]$ respectively.

Algorithm 2 Comparison of item importance rankings

```
  **Input** Item sequence $\langle i_1, i_2, \dots i_n \rangle$, target item $i^*$.
  **Output** Rank correlation of item importance metrics.
  $\langle \mathbf{h}_1, \mathbf{h}_2, \dots, \mathbf{h}_n \rangle,\ \langle \mathbf{A}^{(1)}, \dots \mathbf{A}^{(b)} \rangle \leftarrow \text{Enc}(\langle i_1, i_2, \dots i_n \rangle)$
  $\nabla_j \leftarrow \|\frac{\partial \mathbf{h}_n^\top \mathbf{e}_{i^*}}{\partial \mathbf{e}_{i_j}}\|_2 \quad \text{for } j = 1, \dots, n$
  $\varphi \leftarrow \text{Shapley\_values}(\langle i_1, i_2, \dots i_n \rangle, i^*)$
  $\alpha_{\text{first}} \leftarrow \mathbf{A}^{(1)}_{n,\cdot}$
  $\alpha_{\text{last}} \leftarrow \mathbf{A}^{(b)}_{n,\cdot}$
  $\alpha \leftarrow \text{attention\_rollout}(\langle \mathbf{A}^{(1)}, \dots \mathbf{A}^{(b)} \rangle)$
  **return** $\rho_{\text{Spearman}}(u, v) \quad \text{for } (u, v) \in \{\alpha_{\text{first}}, \alpha_{\text{last}}, \alpha\} \times \{\delta, |\varphi|\}$
```

The Spearman rank correlation is a popular choice in the explainable AI literature to compare the agreement of importance rankings produced by different explainability methods, e.g. [1, 4, 45], for the following reason: The scales on which importance scores by different methods are computed will typically not be directly comparable, hence the Pearson correlation between scores would not be meaningful. Instead, Spearman rank correlation allows us to compare the ordering of importance scores, i.e. to measure if important items according to one ranking are also important according to the other. We therefore adapt the Spearman rank correlation as a metric to compare importance rankings by attention scores, analogous to how [25] employ the less common Kendall rank correlation coefficient.

3.4.2 Discounted cumulative gain

[25] note that a potential issue with rank correlation is that a potentially large number of irrelevant items may add noise and that as a result the correlation coefficient may be fairly low, despite attention and post-hoc methods mostly agreeing on which items they deem the most important. In this section propose a novel metric designed to alleviate this issue.

We wish to measure the agreement of the ranks r_{X_i}, r_{Y_i} from a sequence of n paired observations as in Section 3.4.1. We define the following version of *discounted cumulative gain* by

$$\mathrm{DCG}(X,Y) = \sum_{0 \le i < n} \frac{n - r_{X_i}}{\log_2 (r_{Y_i} + 2)}.$$

Sematically, the idea is that $n - r_{X_i}$ serves as a “ground-truth” relevance of item i, i.e. the item with the lowest rank r_{X_i} has the highest relevance and relevance decreases linearly with rank. $\mathrm{DCG}(X,Y)$ then discounts the relevance scores logarithmically according the “predicted” ranks r_{Y_i}, meaning that placing an item twice as far back in the r_{Y_i} ranking halves the discounted relevance. Consequently, the discounted cumulative gain places an emphasis on r_{Y_i} getting the top ranks right.

To make the values of the discounted cumulative gain comparable across different sequence lengths n, we consider a normalised version

$$\text{N-DCG}(X,Y) = 2\left(\frac{\mathrm{DCG}(X,Y) - \mathrm{DCG}_{\min}(n)}{\mathrm{DCG}_{\max}(n) - \mathrm{DCG}_{\min}(n)}\right) - 1$$

where

$$\mathrm{DCG}_{\max}(n) = \mathrm{DCG}(\langle 1,2,\ldots,n\rangle, \langle 1,2,\ldots,n\rangle),$$

$$\mathrm{DCG}_{\min}(n) = \mathrm{DCG}(\langle 1,2,\ldots,n\rangle, \langle n,n-1,\ldots,1\rangle)$$

are the largest and smallest discounted cumulative gain scores possible for sequences of length n, respectively. Thus, $\text{N-DCG}(X,Y)$ takes values in $[-1,1]$ and for independant X,Y we have $\mathbb{E}[\text{N-DCG}(X,Y)] = 0$

Since $\mathrm{DCG}(X,Y)$ and consequently also $\text{N-DCG}(X,Y)$ are asymmetric, we derive (in the style of the Jensen-Shannon divergence) the symmetric normalised discounted cumulative gain

$$\mu_{\log}(X,Y) = \frac{1}{2}\text{N-DCG}(X,Y) + \frac{1}{2}\text{N-DCG}(Y,X)$$

as a symmetric measure for rank agreement, emphasising agreement on the top ranks.

Unfortunately, it is not clear how our new metric relates to the Spearman rank correlation. Therefore, we cannot directly use it to detect if the agreement between attention scores and post-hoc methods is stronger on important items. Thus, we define an alternative version of discounted cumulative gain

$$\text{DCG}_{\text{lin}}(X,Y) = \sum_{0 \le i < n} \frac{n - r_{X_i}}{(r_{Y_i} + 2)}.$$

with relevance decaying linearly, instead of logarithmically.[3] As a result DCG_{lin} places even stronger emphasis on the top ranks than DCG. We define $\mu_{\text{lin}}(X,Y)$ analogously to $\mu_{\log}(X,Y)$. The point is that we can now compare μ_{lin} to $\mu_{\log}$ scores to measure if the rankings agree more towards the top.

This is best illustrated by an example: If the sequences disagree on their bottom ranks, the top-heavier μ_{lin} score is greater than the $\mu_{\log}$ score:

$$\mu_{\log}(\langle 1,2,\mathbf{3},\mathbf{4}\rangle, \langle 1,2,\mathbf{4},\mathbf{3}\rangle) \approx 0.891 \le 0.898 \approx \mu_{\text{lin}}(\langle 1,2,\mathbf{3},\mathbf{4}\rangle, \langle 1,2,\mathbf{4},\mathbf{3}\rangle).$$

However, if the sequences disagree on their top ranks, the μ_{lin} score is less than the $\mu_{\log}$ score:

$$\mu_{\log}(\langle \mathbf{1},\mathbf{2},3,4\rangle, \langle \mathbf{2},\mathbf{1},3,4\rangle) \approx 0.673 \ge 0.661 \approx \mu_{\text{lin}}(\langle \mathbf{1},\mathbf{2},3,4\rangle, \langle \mathbf{2},\mathbf{1},3,4\rangle).$$

While the difference in scores is small, in our experiments in Section 4.2.2 it is statistically significant, and thus sufficient to provide us with evidence in favour of the conjecture of [25] mentioned at the beginning of this section.

3.5 Further metrics

3.5.1 Jensen–Shannon divergence

The *Kullback–Leibler divergence* (or *relative entropy*) of two discrete probability distributions $p = \langle p_k : k < k^* \rangle$, $q = \langle q_k : k < k^* \rangle$ is defined as

$$\text{KL}(P \| Q) = \sum_{k<k^*} p_k \log(\frac{p_k}{q_k}).$$

[3] It does not matter that relevance decays exactly linearly, any function that grows faster than the logarithm would work.

It is an asymmetric divergence metric rooted in information theory. To obtain a symmetric divergence metric, the Jensen-Shannon divergence is defined as

$$\mathrm{JS}(P,Q) = \frac{\mathrm{KL}(P\|Q) + \mathrm{KL}(Q\|P)}{2}.$$

Similar to [46], we use the Jensen-Shannon divergence to measure the change in the probability distribution over the predicted next item in our deletion experiment in Section 4.2.1.

3.5.2 Jaccard index

The Jaccard index measures overlap between finite sets. It is defined as

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|}.$$

As an addition to the tools described in section 3.4.2 we use it in Section 4.2.2 to show agreement between the top items of different important rankings.

Chapter 4

Results

4.1 Main experiment: rank correlation

We focus on the description of our results in this section, while leaving their interpretation mostly for Section 5. We accompany most tables by a graphical duplication of their contents to allow for quick visual assessment of general trends in the numeric results.

4.1.1 Single attention block

We begin by investigating the simplest case of Transformers with a single attention block ($b = 1$) and with a single attention head ($h = 1$) per block. The resulting summary statistics are reported in Table 4.1 and Figure 4.1.

4.1.1.1 Gradients

SASRec. For the SASRec model we observe that attention is strongly correlated with the gradients for the ML-1M and Steam datasets and moderately correlated for the Beauty and Games datasets.[1] On the other hand, we obtain near perfect correlation between the attention rollout (AR) scores and the gradients for all datasets. For a single attention block the difference between attention and AR is that AR takes into account the skip connections in the transformer. This suggests that even for a single attention block it is necessary to be mindful of the model's architecture when attempting to construct intrinsic explanations from attention scores.

[1] To some extend correlation between attention scores of the first attention block and gradients is not surprising, since a larger attention score for an item makes the model more susceptible to sensitivity to this item. However, it is not clear how strong this correlation "should" be.

Model	Dataset	$\rho(\nabla, \alpha_{\text{first}})$	$\rho(\nabla, \alpha)$	$\rho(\lvert\varphi\rvert, \alpha_{\text{first}})$	$\rho(\lvert\varphi\rvert, \alpha)$
SASRec	ML-1M	0.97 ± 0.04	0.98 ± 0.02	0.47 ± 0.15	0.48 ± 0.15
	Steam	0.84 ± 0.27	0.98 ± 0.06	0.30 ± 0.34	0.27 ± 0.33
	Beauty	0.60 ± 0.44	0.99 ± 0.04	0.26 ± 0.47	0.12 ± 0.48
	Games	0.61 ± 0.43	0.99 ± 0.02	0.11 ± 0.45	0.16 ± 0.44
BERT4Rec	ML-1M	0.99 ± 0.01	0.99 ± 0.01	0.53 ± 0.11	0.53 ± 0.11
	Steam	0.97 ± 0.08	0.97 ± 0.08	0.58 ± 0.29	0.58 ± 0.29
	Beauty	0.97 ± 0.08	0.97 ± 0.08	0.49 ± 0.41	0.49 ± 0.41
	Games	0.96 ± 0.09	0.96 ± 0.09	0.48 ± 0.43	0.48 ± 0.43

Table 4.1: $b = 1$ attention blocks. Spearman-ρ rank correlation between gradients ∇, Shapley values φ, attention scores α_{first}, and attention rollout scores α. Mean ± standard deviation for 1000 data points. Note: the standard deviation measures how spread out the distributions in Figure 4.2 are, i.e. it is not the standard error of the mean estimate (which is obtained from the standard deviation by dividing the value by $\sqrt{1000}$).

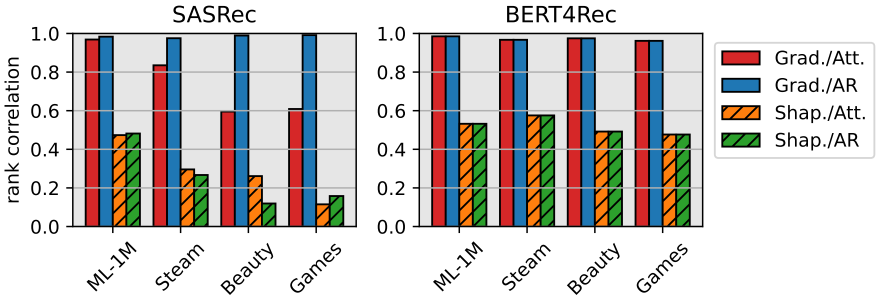

Figure 4.1: $b = 1$ attention blocks. Visualisation of Table 4.1.

BERT4Rec. For the BERT4Rec model we observe consistently high correlation between attention and gradients for all datasets. Remember that for BERT4Rec the attention and AR scores are equal for $b = 1$ since we omit the scores associated with the `[CLS]` token, as discussed in Section 3.3.2. Therefore, of course, also the correlations between attention and gradients are equal to the correlations between AR scores and gradients.

4.1.1.2 Shapley values

SASRec. For the SASRec model we observe moderate correlation between the absolute Shapley values and the attention scores for the ML-1M dataset and weak correlation for the remaining datasets. Unlike for the gradients, using the AR scores does not help here: the correlation only improves marginally for the ML-1M dataset and even

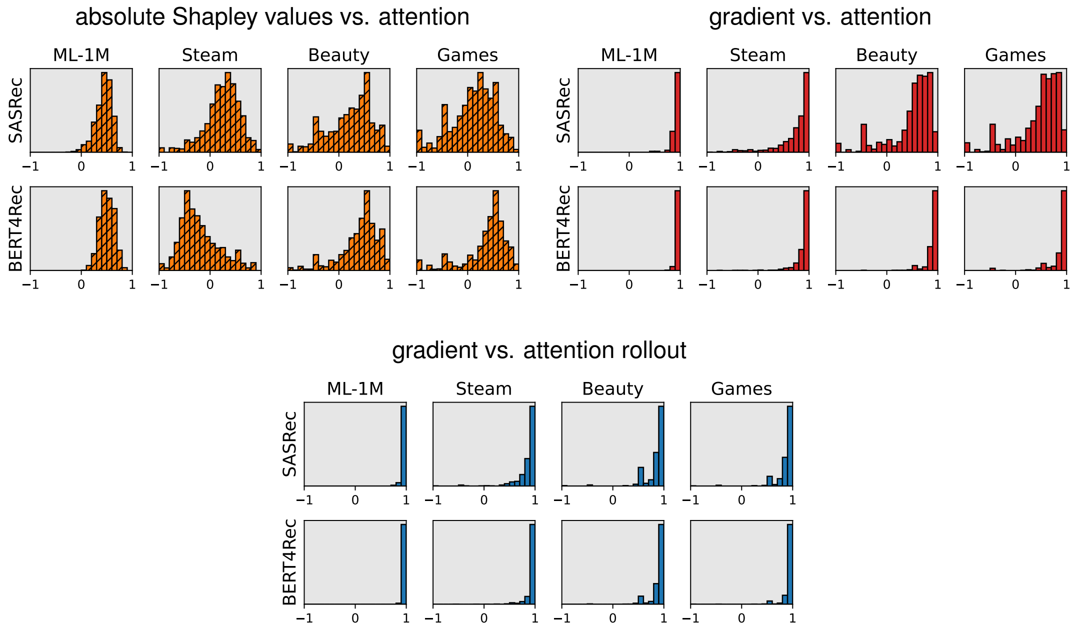

Figure 4.2: Histograms of the distributions of the Spearman rank correlations between importance rankings, estimated from 1000 data points. For each histogram, the bars sum up to 100% (i.e. the y-axes are different scales). The x-axis corresponds to the Spearman rank correlation.

decreases for the remaining datasets.

BERT4Rec. For the BERT4Rec model we observe moderate correlation between the absolute Shapley value and the attention (respectively AR) scores.

To summarise: For either model type and all datasets we observe near perfect rank correlations between the gradient-based importance ranking and the AR scores. In particular, we saw that AR can be helpful, even for a single attention block.

For the absolute Shapley values we only observe weak to moderate correlations with attention. We should however keep in mind that, despite plotting correlations on a scale $[0,1]$, the Spearman rank correlation takes values in $[-1,1]$ and we consistently observe positive values, so it would be incorrect to conclude that attention scores and Shapley values are completely unrelated.

4.1.1.3 Distribution of correlations

Figure 4.2 shows the distribution histograms of the correlations for gradients and attention respectively AR, and absolute Shapley values and attention. We observe that for the latter, for the ML-1M dataset of either model the distribution is somewhat tightly peaked in the positive region, while for the other datasets the distribution is more spread out, with a considerable portion of mass in the negative region.

On the other hand, for the the correlations between gradients and attention scores, the distributions have most of their mass close to 1. For the SASRec model, the distributions are slightly more spread out for the Steam dataset than for the ML-1M dataset, and considerably more spread out for the Beauty and Games datasets. For the correlations between gradients and AR scores, the distributions are concentrated even more towards 1 for for all datasets and models, even without any outliers for the ML-1M set.

We note that for all three correlations types, the ML-1M dataset is the “best behaved” (in the sense that the distribution is clustered around its mean), the sparser Steam dataset is slightly worse behaved and the sparsest Beauty and Games datasets are the worst behaved. A possible explanation could be that for denser datasets the model sees items in more different contexts and can learn more informative attention scores that correlate better with other importance measures.

4.1.2 Multiple attention blocks

Next, we investigate the case of multiple attention blocks, again with $h = 1$ heads per block. The results are reported in Table 4.2 and Figure 4.3 for the case of $b = 2$ attention blocks, and in Table 4.3 and Figure 4.4 for the case of $b = 3$ attention blocks.

4.1.2.1 Gradients

SASrec. For the SASrec models with $b = 2,\ b = 3$, we observe that the attention scores of the first attention block correlate strongly with the gradients for the ML-1M and Steam datasets, and moderately for Games and Beauty. These correlations are all slightly weaker than the corresponding correlations for $b = 1$. Again, the AR scores correlate much more strongly with the gradients than the attention scores of the first block. Notably, for $b = 3$ and the Beauty and Games datasets AR correlates only moderately with the gradients, while in all other cases the correlation is strong. The correlations between attention scores of the last block are consistently lower than those of the first block.

BERT4Rec. For the BERT4Rec model, the correlations between the attention scores of the first block and gradients are consistently high, with the AR still slightly improving correlations, except for the Beauty dataset where for $b = 3$ AR very slightly decreases correlation. The correlations between the attention scores in the last block with the gradients are consistently lower than those for the first block, except for the

Model	Dataset	$\rho(\nabla, \alpha_{\text{first}})$	$\rho(\nabla, \alpha_{\text{last}})$	$\rho(\nabla, \alpha)$	$\rho(\lvert\varphi\rvert, \alpha_{\text{first}})$	$\rho(\lvert\varphi\rvert, \alpha)$
SASRec	ML-1M	0.94 ±0.06	0.88 ±0.08	0.98 ±0.03	0.44 ±0.16	0.45 ±0.16
	Steam	0.70 ±0.29	0.63 ±0.29	0.90 ±0.14	0.25 ±0.35	0.25 ±0.35
	Beauty	0.54 ±0.42	0.33 ±0.47	0.86 ±0.23	0.24 ±0.46	0.17 ±0.48
	Games	0.44 ±0.45	0.19 ±0.48	0.87 ±0.19	0.13 ±0.45	0.07 ±0.44
BERT4Rec	ML-1M	0.95 ±0.04	0.81 ±0.12	0.97 ±0.03	0.50 ±0.14	0.50 ±0.14
	Steam	0.87 ±0.16	0.90 ±0.16	0.96 ±0.08	0.49 ±0.31	0.47 ±0.33
	Beauty	0.93 ±0.12	0.52 ±0.42	0.94 ±0.12	0.45 ±0.40	0.45 ±0.41
	Games	0.90 ±0.20	0.86 ±0.18	0.95 ±0.13	0.37 ±0.44	0.40 ±0.43

Table 4.2: $b = 2$ attention blocks. Spearman-ρ rank correlation between gradients ∇, Shapley values φ, attention scores α_{first},α_{last}, and attention rollout scores α. Mean ± standard deviation for 1000 data points.

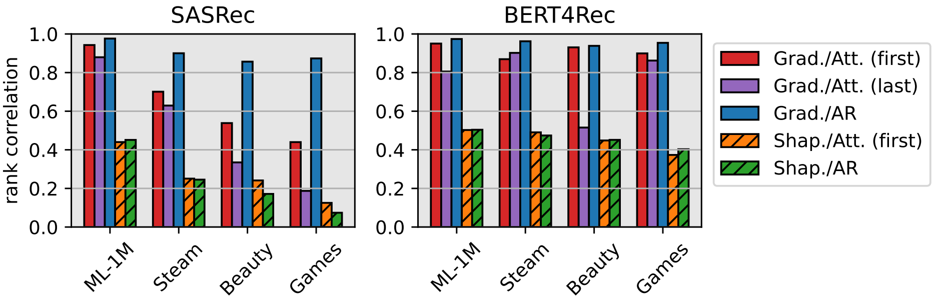

Figure 4.3: $b = 2$ attention blocks. Visualisation of Table 4.2.

Steam dataset for $b = 2$ where the correlation is very slightly stronger.

4.1.2.2 Shapley values

SASrec. For the SASRec model, the correlations between the attention scores and the absolute Shapley values are again only moderate for ML-1M dataset and lower for the remaining datasets. Again, AR does little to improve correlations for Shapley values.

BERT4Rec. As for $b = 1$, for the BERT4Rec model the correlations between absolute Shapley values and attention scores are moderate. Once again, AR scores do not help.

To summarise: For either model type and all datasets we observe strong rank correlations between the gradient based importance scores and the AR scores, except for the SASRec model with $b = 3$ for the Beauty and Games datasets where the correlations were only moderate (but still stronger than any other importance measure considered). With one minor exception, the correlations between the attention scores of the last

Model	Dataset	$\rho(\nabla, \alpha_{\text{first}})$	$\rho(\nabla, \alpha_{\text{last}})$	$\rho(\nabla, \alpha)$	$\rho(\lvert\varphi\rvert, \alpha_{\text{first}})$	$\rho(\lvert\varphi\rvert, \alpha)$
SASRec	ML-1M	0.89 ±0.12	0.75 ±0.15	0.97 ±0.03	0.38 ±0.15	0.41 ±0.14
	Steam	0.63 ±0.33	0.43 ±0.38	0.91 ±0.13	0.27 ±0.35	0.30 ±0.33
	Beauty	0.59 ±0.38	0.28 ±0.45	0.71 ±0.38	0.30 ±0.45	0.30 ±0.44
	Games	0.46 ±0.43	0.21 ±0.47	0.71 ±0.37	0.06 ±0.46	0.13 ±0.44
BERT4Rec	ML-1M	0.93 ±0.06	0.85 ±0.11	0.97 ±0.04	0.37 ±0.14	0.40 ±0.14
	Steam	0.87 ±0.19	0.80 ±0.27	0.94 ±0.12	0.46 ±0.31	0.45 ±0.32
	Beauty	0.95 ±0.11	0.64 ±0.38	0.92 ±0.14	0.34 ±0.46	0.35 ±0.46
	Games	0.87 ±0.22	0.65 ±0.42	0.94 ±0.12	0.30 ±0.46	0.34 ±0.45

Table 4.3: $b = 3$ attention blocks. Spearman-ρ rank correlation between gradients ∇, Shapley values φ, attention scores α_{first},α_{last}, and attention rollout scores α. Mean ± standard deviation for 1000 data points.

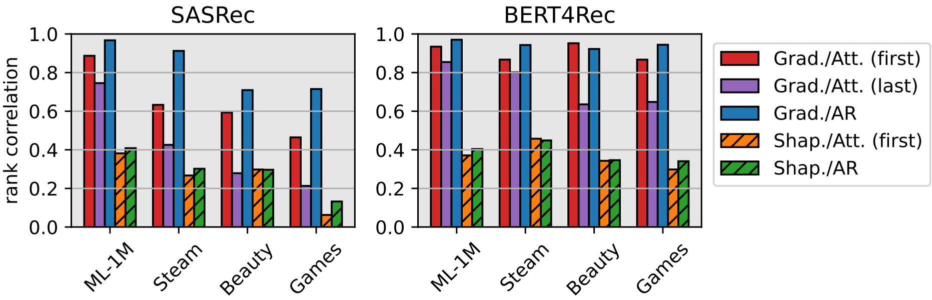

Figure 4.4: $b = 3$ attention blocks. Visualisation of Table 4.3.

block and the gradients were lower (and often much lower) than those for the first attention block.

The correlations between the absolute Shapley values and the attention scores were again much weaker than those for the gradients, but still strictly non-negative.

4.1.3 Multiple attention heads

Finally, we investigate the effect of using multiple attention heads. For $b = 2$ attention blocks and $h = 2$ heads per attention block, the results are reported in Tabel 4.4 and Figure 4.5.

We consider the attention scores for each of the two attention heads, as well as the mean of the two. We observe that for each individual attention head the correlations between attention scores and and the gradients are either slightly weaker or about the same as for the mean attention scores.

SASRec. For the SASRec model, the correlations between mean attention scores

M.	Dataset	$\rho(\nabla, \alpha_{\text{first},1})$	$\rho(\nabla, \alpha_{\text{first},2})$	$\rho(\nabla, \alpha_{\text{first}})$	$\rho(\nabla, \alpha)$	$\rho(\alpha_{\text{first},1}, \alpha_{\text{first},2})$
SASRec	ML-1M	0.87 ±0.10	0.87 ±0.10	0.94 ±0.07	0.97 ±0.03	0.76 ±0.16
	Steam	0.60 ±0.30	0.70 ±0.24	0.71 ±0.24	0.88 ±0.18	0.77 ±0.31
	Beauty	0.71 ±0.31	0.72 ±0.31	0.72 ±0.31	0.93 ±0.15	0.96 ±0.10
	Games	0.55 ±0.39	0.54 ±0.40	0.55 ±0.39	0.91 ±0.15	0.93 ±0.15
BERT4Rec	ML-1M	0.89 ±0.06	0.76 ±0.13	0.90 ±0.07	0.96 ±0.03	0.74 ±0.10
	Steam	0.68 ±0.35	0.69 ±0.27	0.90 ±0.18	0.95 ±0.11	0.24 ±0.45
	Beauty	0.90 ±0.19	0.90 ±0.20	0.91 ±0.19	0.97 ±0.08	0.97 ±0.07
	Games	0.78 ±0.26	0.82 ±0.23	0.82 ±0.22	0.94 ±0.14	0.89 ±0.18

Table 4.4: $b = 2$ attention blocks with $h = 2$ heads each. Spearman-ρ rank correlation between gradients ∇, attention scores for the first and second head $\alpha_{\text{first},1}$ and $\alpha_{\text{first},2}$ respectively, the mean attention scores α_{first} over both heads, and attention rollout scores α. Mean ± standard deviation for 1000 data points.

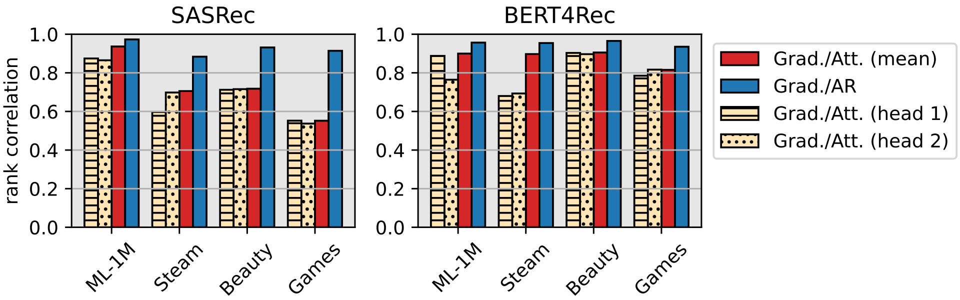

Figure 4.5: $b = 2$ attention blocks with $h = 2$ heads each. Visualisation of Table 4.4.

and gradients are strong for the ML-1M dataset and moderate for the remaining datatsets. For the attention rollout they are strong for all datasets.

BERT4Rec. For the BERT4Rec model, correlations between mean attention scores and gradients are strong for all datasets, and slightly increase further for the attention rollout.

To summarise: the attention rollout algorithm handles multiple attention heads well, resulting in strong rank correlations with the gradient-based importance ranking in all cases.

Furthermore, it is worth noting that the attention scores for the two heads are strongly correlated with each other for all models except the BERT4Rec model for the Steam dataset. We find this not surprising since as we saw in Section 3.2.4 the additional heads do not add performance, i.e. the model appears not to utilise their availability.

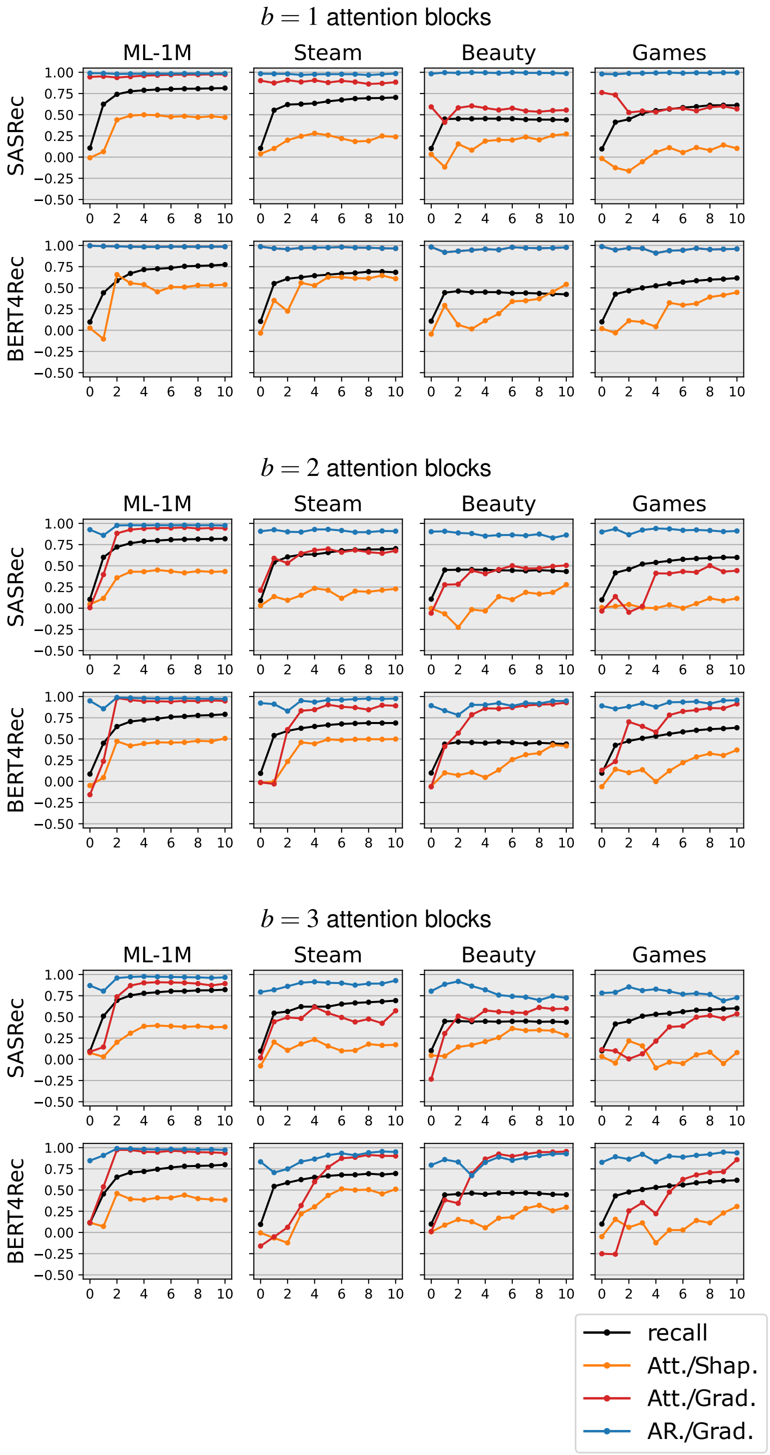

Figure 4.6: Training dynamics over 10 epochs (x-axis), as described in Section 3.2.4. Spearman rank correlations of the importance metrics attention, attention rollout, gradients and absolute Shapley values, as well as the recall@10% metric.

4.1.4 Training dynamics

Figure 4.6 shows the Spearman rank correlations between attention scores and absolute shapley values, respectively gradients, as well as AR scores and gradients as they change during training. Furthermore, the recall@10% metric is shown.

We observe that the correlations between AR scores and gradients are the most stable, usually remaining high throughout training. The correlations involving the raw attention scores are much less stable both for gradients and Shapley values. They sometimes dip below 0 at the beginning of training and only noisily rise throughout training.

Something else worth pointing out is that even in cases where the correlation between Shapley values and attention is well behaved and increases more or less monotonically during training (the SASRec models with $b = 1,2,3$ and the ML-1M dataset) the knee of this curve is behind the knee of the recall@10 curve, i.e. the model performs fairly well while the correlation remains low, and only then the correlation rises. We cannot offer an explanation for this phenomenon.

Overall, we consider the results regarding the stability of the AR scores to be additional evidence that these scores are a promising method to construct item importance rankings from attention scores.

4.2 Auxiliary experiments

4.2.1 Item deletion

In this section we experiment with deleting the most important items according to different importance rankings. We measure the Jensen–Shannon divergence of the resulting probability distribution for the next item compared to the distribution with all items present. This experiment is similar to one conducted in [46]. However, there the authors measure how many items need to be deleted on average for the model's decision to "flip", i.e. for the item with the highest predicted probability to change. [46] conduct this experiment for text classification problems with 5 or 10 classes, while we predict probabilities over thousands of items. Therefore (and also because a sequence may result in several reasonable recommended items) the exact top item is less meaningful and we choose to work with the Jensen–Shannon divergence instead. The intuition however remains the same: the most important items should induce the strongest change in the prediction. Thus, by comparing how quickly predictions change for

	ML-1M	Steam	Beauty	Games
SASRec	4.41 ±0.45	3.71 ±1.02	2.58 ±1.05	2.98 ±1.05
BERT4Rec	3.97 ±0.23	3.57 ±0.95	2.50 ±0.93	2.72 ±0.85

Table 4.5: Mean entropy of $\alpha_{\text{first}} \pm$ standard deviation for 1000 datapoints.

different importance rankings allows us to compare their explanatory power.

The resulting divergences, for $b = 2$ attention blocks with $h = 1$ heads each, are shown in Figure 4.7. For all models, we observe that deletion of items in random order leads to the slowest changes in the probability distribution, essentially showing that all considered ranking methods have more explanatory power than a random baseline. For the SASrec models, the absolute Shapley values are between the other methods and the random ordering in explatory power, while for the BERT4Rec model they are actually very slightly above the other methods. For all model types, attention scores, attention rollout and gradients have similar explanatory powers (which is in part explained by the results of the previous section showing that these metrics are positively correlated).

However, the most striking observation is that for the SASRec model the deletion curves are convex (except for some rankings in the ML-1M dataset) while for the BERT4Rec model they are concave (or approximately concave with the curves for the ML-1M dataset "over-shooting" and even becoming non-monotonic). While we introduced explainable AI from the perspective of explanations for the user, there is also the aspect of understanding models from the point of the engineer. Our observation regarding the shapes of the curves thus allows us the following conjecture regarding the workings of our models: the SASrec models base their predictions fairly uniformly on items in the input sequence, while the BERT4Rec models base their decision on few, very important items, hence deleting these items quickly results in change of the prediction. This conjecture is further supported by the entropy of the attention scores reported in Table 4.5. The entropy of the attention scores for the SASRec models is consistently higher than the entropy for the BERT4Rec models on the same dataset.

Overall, we consider the results in this section to be additional support for attention as explanation. Attention had explanatory power consistently higher than that of the Shapley values and roughly equal to that of the gradients for the SASRec models, and roughly equal for either for the BERT4Rec models. This finding is different from the results of [46] where the authors find that gradients have higher explanatory power than attention for text classification, when measured by decision flips.

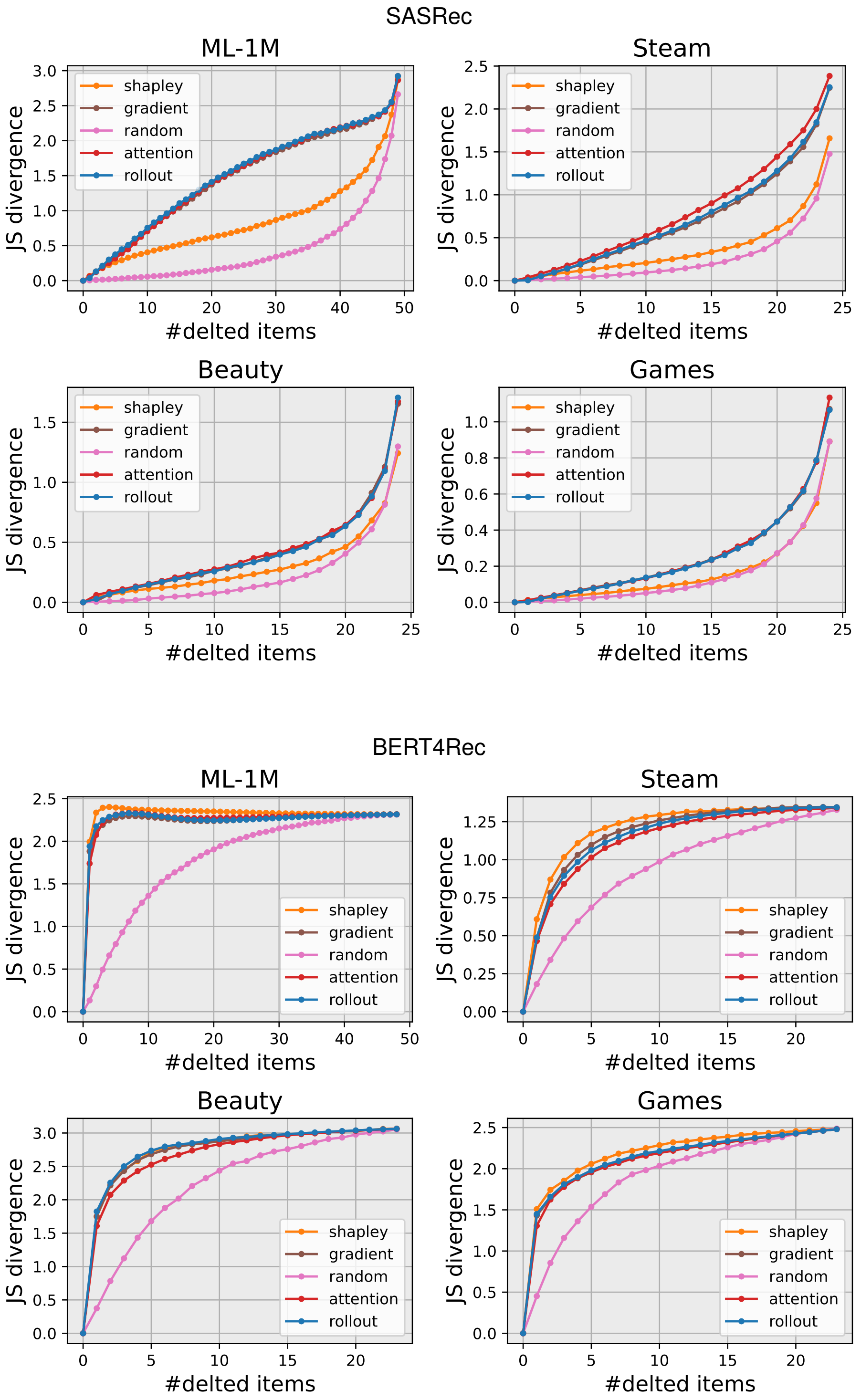

Figure 4.7: Change in Jensen–Shannon divergence as a function of the number of deleted items. The first item deleted is the most important item according to the respective importance ranking. Mean over 500 input sequences with full length (50 items for ML-1M, 25 items for the remaining datasets).

M.	Dataset	$\rho(\lvert\varphi\rvert,\alpha_{\text{first}})$	$\mu_{\text{log}}(\lvert\varphi\rvert,\alpha_{\text{first}})$	$\mu_{\text{lin}}(\lvert\varphi\rvert,\alpha_{\text{first}})$	p	$J_3(\alpha_{\text{first}},\lvert\varphi\rvert)$
SASRec	ML-1M	0.47 ±0.15	0.609 ±0.17	0.635 ±0.19	6.3e-107	0.08 ±0.12
	Steam	0.30 ±0.34	0.351 ±0.38	0.359 ±0.39	4.8e-23	0.28 ±0.31
	Beauty	0.26 ±0.47	0.296 ±0.50	0.300 ±0.50	1.3e-11	0.50 ±0.36
	Games	0.11 ±0.45	0.146 ±0.48	0.151 ±0.49	1.2e-07	0.39 ±0.35
BERT4Rec	ML-1M	0.53 ±0.11	0.781 ±0.07	0.826 ±0.07	3.0e-164	0.07 ±0.13
	Steam	0.58 ±0.29	0.642 ±0.32	0.650 ±0.32	1.6e-43	0.34 ±0.31
	Beauty	0.49 ±0.41	0.536 ±0.43	0.541 ±0.43	5.2e-20	0.55 ±0.36
	Games	0.48 ±0.43	0.529 ±0.46	0.535 ±0.46	6.3e-27	0.49 ±0.36

Table 4.6: The Spearman-ρ rank correlation, as well as the symmetric normalised discounted cumulative gains (μ_{log},μ_{lin}) between Shapley values φ and attention scores α_{first}, for $b = 1$ attention blocks, as well as the Jaccard index J_3 for the top 3 items according to $|\varphi|$ and α_{first}. Mean ± standard deviation for 1000 data points. The "p" column reports the p-value of the Wilcoxon signed-rank test for $\mu_{\text{log}} < \mu_{\text{lin}}$.

4.2.2 Symmetric normalised DCG for Shapley values

[26] conjecture that correlations between importance rankings may be low due to noise resulting from a large number of irrelevant items, despite the rankings mostly agreeing on the top items. In this section we utilise our novel symmetric normalised discounted cumulative gain metrics to find evidence for this conjecture for the case of the correlation between absolute Shapley values and attention. Table 4.6 shows the measurements of these metrics for $b = 1$ attention blocks. As we discussed in Section 3.4.2, despite μ_{log} weighing top items more strongly, we cannot directly compare this score to the Spearman-ρ, so instead we compare it to the even top-heavier μ_{lin}. For all experiments, the μ_{log} scores between the absolute Shapley values and the attention weights are lower than the corresponding μ_{lin} scores. While the absolute differences between these measurements are small (remember the example from Section 3.4.2 where the absolute differences were also small), they are statistically significant, according to the Wilcoxon signed-rank test. We consider this evidence in favor of the conjecture of [25], for the case of Shapley values and attention scores.

An additional piece of evidence is provided by the Jaccard indices for the top 3 items according to the absolute Shapley values and the attention scores, reported in Table 4.6. Note that these scores are higher than they appear: if 2 out of the top 3 items agree the Jaccard index is $\frac{2}{4} = 0.5$ and if 1 out of the top 3 items agree it is $\frac{1}{5} = 0.2$.

Thus, except for the ML-1M dataset, the Jaccard indices suggest that there is indeed a substantial agreement for the top 3 items.

4.2.3 Movie genres

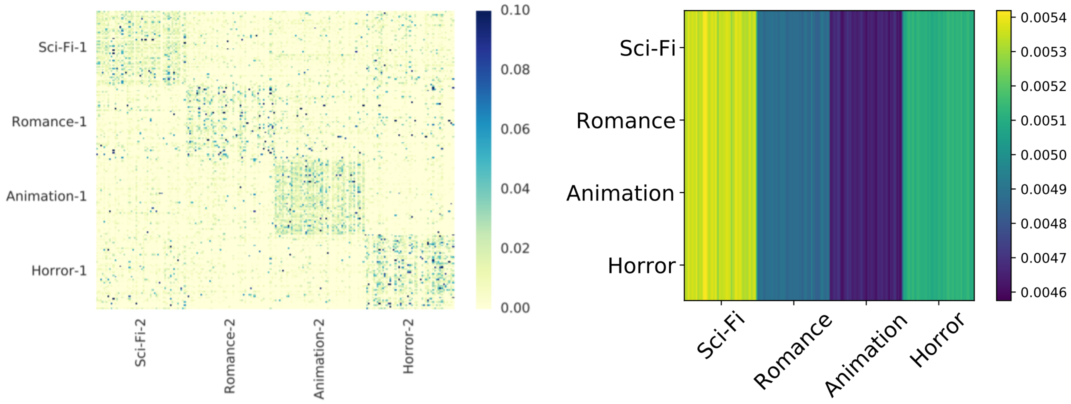

Figure 4.8: Left: Figure from [26]. Right: our reproduction: Attention between items for 10,000 pairs of random item sequences from the ML-1M dataset of length 200, with 50 items each taken from 4 selected genres.

We attempt to reproduce an experiment by [26]. For the ML-1M dataset we generate pairs of random sequences of 200 movies, with 50 movies each from the categories "Science Fiction" "Romance", "Animation", and "Horror" (in this order, e.g. the first 50 movies of each sequence are science fiction movies). Figure 4.8 shows the resulting average attention for 10,000 sequence pairs, taking the first sequence as query and the second as key, for the SASRec model with $b = 1$ attention blocks. Unlike [26] who observe roughly a block diagonal matrix, we observe "stripes".

While we failed to reproduce the result of [26][2], the conclusion we draw from our result is very much the same as theirs: despite never being told about movie genres, a pattern related to movie genres appears in the attention scores of the model. Regarding the question whether attention is explanation, we consider this result to be in favour of attention as explanation, since the attention scores appear to have picked up a semantically meaningful pattern.

[2] Their description of this experiment is very coarse, it is possible we misunderstood their set up. The z-scales of Figure 4.8 also suggests that the authors may have conducted an experiment different from ours: the range of values they observe are much higher than the values of $\approx \frac{1}{200}$ we observe, which are reasonable based on our experimental setup.

Chapter 5

Discussion

5.1 Insights gained

In this section we summarise the insights gained through our experiments in Section 4.1.

Attention over intermediate representation may be problematic. As pointed out by [39], many works on attention as explanation overlook the fact that they use attention over intermediate representations that contain information not from more than just a single item of the input sequence. According to our experiments, this may be a significant oversight. In Section 4.1.2 we saw that the correlation between the gradient-based importance ranking and the attention-based rankings were consistently weaker for the intermediate representations of last attention block than than for the first block (and often much weaker).

Model architecture matters. Even when considering models with a single attention block in Section 4.1.1, the correlation with the gradient-based importance rankings increased considerably for the SASRec models when taking into account the skip-connections of the transformer architecture through a single block attention rollout, instead of naively taking raw attention scores as explanation.

Attention should be traced through the model. This insight is a combination of elements of the previous points. Rather than taking raw attention scores, it is beneficial to trace the attention through the model according to its architecture by employing the attention rollout algorithm. In Section 4.1.2 we saw that this usually drastically increases the correlation with the gradient-based importance ranking.

Overall, our results contradict the concern of [25], stating "(...) that attention consistently correlates poorly with multiple such (importance) measures ought to give pause to practitioners” which leads them to their conclusion that “attention is not explanation”. Instead, our experiments suggest that the approach of employing raw attention scores (in particular, over intermediate representations) as explanation may be too naive. Employing the attention rollout algorithm allowed us to construct an importance measure from attention scores with consistently moderate (and in almost all cases strong) correlation with the gradient-based importance ranking. In particular, while [25] continue to state that “(...) exactly how strong such correlations ‘should’ be in order to establish reliability as explanation is an admittedly subjective question” we consider the correlations we observed strong enough to dispel such concerns. Thus, if we had to summarise our results into a single sentence we would conclude: While raw attention may not be explanation, the attention rollout algorithm appears to be a promising tool to construct an importance ranking from attention scores that may serve as explanation for sequential recommender systems.

5.2 Limitations

Our work studies the relationship between attention and established feature importance rankings from the explainable AI literature. What our work is not, is an instruction how to successfully build a sequential recommender system with attention-based explainability. Rather, it should be considered a preliminary study, whether building such system could be in principal possible or if attention is indeed not explanation as suggested by previous work. In Section 5.3 we are briefly going to discuss what such system could look like and how one would evaluate it.

It is worth pointing out that our results are purely empirical (just like the results of [25, 60, 46, 39] etc.). However, since we chose several different datasets of varying domains and characteristics and different model types for our experiments, we find it reasonable to hope that our results reflect general trends. Furthermore, since the attention rollout algorithm is reminiscent of *layerwise relevancy propagation*, an explainability method that has been shown to simply result in gradient times input for ReLU networks [49], one may wonder if the attention rollout scores being strongly correlated with gradient-based importance is also a mathematical necessity. However, our experiments show that it is not, since for the SASRec model with $b = 3$ attention blocks we only obtained moderate correlations for two datasets.

As a student project, our work was limited by our computational budget. In order to try different model types, configurations and datasets in various combinations we

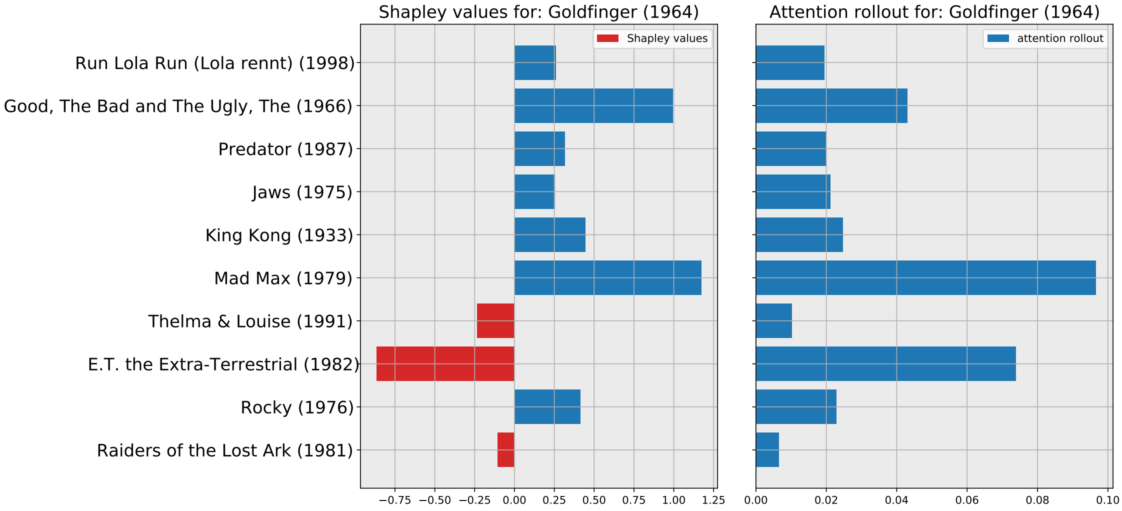

Figure 5.1: Example explanations for the SASRec model with $b = 2$ attention blocks for the ML-1M dataset. Based on the sequence of movies listed on the y-axis (with the most recently watched movie at the top) the model predicts the movie “Goldfinger”. Left: Shapley values for each movie for this prediction. Right: attention rollout scores for this prediction.

had to limit ourselves to one run per experiment. While this is surprisingly common in the recommender systems literature [11] this is of course not ideal.

Finally, it is worth reiterating that in this dissertation we consider the domain of recommender systems but tend to relate our work to results from the domain of natural language processing. To avoid tiresome disclaimers throughout the text we do not mention this explicitly every time, however it should be pointed out that it is possible that our results may not be directly transferable to this domain. In particular, our results in Section 4.2.1 suggest that for certain aspects the domain may indeed make a difference: we found that attention scores and gradients have similar explanatory power according to the change of predictions resulting from item deletions, while [46] found gradients to outperform attention according to a similar metric.

5.3 Outlook

Figure 5.1 gives an example of possible explanations for recommendations for the ML-1M dataset. Shown are importance rankings for the 10 most recently watched movies of a user (out of a total of 50 on which the recommendation is based) generated by the SASRec model with $b = 2$ attention blocks. The figure reiterates the point we made

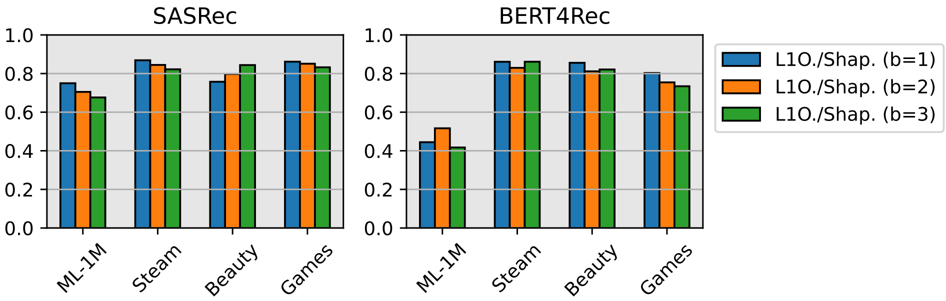

Figure 5.2: Mean Spearman rank correlation between leave-one-out estimates (L1O) and Shapley values, for $b = 1,2,3$ attention blocks with $h = 1$ heads.

in Section 3.3.3 that Shapley values are a "signed" importance ranking differentiating between positive and negative contributions of items. According to the Shapley values users who watch the movie "E.T. the Extra-Terrestrial" are unlikely to watch the predicted movie "Goldfinger", while the attention rollout scores agree that "E.T. the Extra-Terrestrial" is a predictive movie but because it is an unsigned importance ranking they do not tell us that it in fact contradicts the prediction.

Since it is customary to base explanations of recommendations only on positive contributions ("because you liked", but never "because you did not like") an important direction for future work to build recommender systems with attention-based explanations is to find ways to "recover" the sign of the attention scores. A potential candidate are *leave-one-out* estimates, that compare how the predicted probabilities change when deleting single items from the input sequence. Since for an input sequence of length n there are only $O(n)$ leave-one-out estimates (which for sensible sequence lengths can even be computed in a single batch on a GPU, i.e. in constant time), leave-one-out estimates are a cheap alternative to Shapley values which require $O(2^n)$ operations to be computed exactly. Figure 5.2 shows that Shapley values and leave-one-out estimates have moderate to high rank correlation in most cases. Therefore, an idea for future work could be to develop a heuristic that picks out items that have high attention-base importance and are also towards the top of the leave-one-out ranking in order to obtain important, positive items.

5.3.1 Evaluation of explainable recommender systems

We conclude this section with a brief discussion of how an explainable recommender system could be evaluated in practice.

We mentioned in Section 1.1.2 that there is no consensus on what constitutes an explanation, hence it is unsurprising that no universally accepted method for evaluating explanations exists either [32]. If we view explanations as a means to a certain end, then one possibility is to evaluate an explainable recommender system in an online experiment by an A/B test [63], as one would for a regular, non-explainable system [58]. E.g. one could compare click-through rate of recommended items when they are presented with or without explanation [64].

An alternative to A/B tests are user studies [47, 63], where one studies the interaction of users with a recommender systems through tools such as questionnaires or focus groups. E.g. users may be presented with different styles of explanations for movie recommendations and be queried which of them would convince them most to see the recommended movie [21]. An obvious drawback of user studies is that they are time consuming and expensive.

Chapter 6

Conclusion

In this dissertation we investigated the suitability of attention as explanation for sequential recommender systems based on Transformers. Our experiments in Section 4.1 challenge the claim of [25] that “attention is not explanation”. We showed that when employing the right tool, the attention rollout algorithm of [1], the situation may not be quite as hopeless for attention as explanation. The attention rollout scores, constructed by tracing raw attention scores through the Transformer model according to its architecture, obtained very strong correlation with a gradient-based item importance ranking in almost all cases, addressing a central concern of [25] who struggled to relate attention to established post-hoc importance ranking methods. In Section 4.2.2 we were able to provide evidence for a conjecture of [25] that such low correlations may be a result of noisy rankings of unimportant items. For the case of absolute Shapley values and attention, our novel symmetric normalised discounted cumulative gain metrics, specifically designed to measure agreement towards the top, suggested that this is indeed the case.

In Section 4.2.1, by deleting items from input sequences in the order of their importance according to different ranking methods and observing the resulting change in predictions, we observed that for our models attention had similar explanatory power to gradient-based rankings, unlike [46] who observed larger explanatory power for the gradients in a similar experiment for natural language classification tasks. This result can be interpreted as suggesting that attention may be particularly suited as explanation for the recommender systems domain.

Finally, in Section 4.2.3 we were able to further strengthen the case for attention as explanation by showing that an attention mechanism was able to pick up patterns reflecting semantically meaningful genre information, despite the model never having access to this information during training.

While for the recommender systems domain there also remains the conceptual challenge that attention is an unsigned importance measure, we hope that our work inspires future work on attention as explanation, eventually leading to the successful construction of sequential recommender systems with attention-based explanations.

Bibliography

[1] Samira Abnar and Willem Zuidema. Quantifying attention flow in transformers. *arXiv preprint arXiv:2005.00928*, 2020.

[2] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

[3] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[4] Thomas Baumhauer, Djordje Slijepcevic, and Matthias Zeppelzauer. Bounded logit attention: Learning to explain image classifiers. *arXiv preprint arXiv:2105.14824*, 2021.

[5] A Charnes, B Golany, M Keane, and J Rousseau. Extremal principle solutions of games in characteristic function form: core, chebychev and shapley value generalizations. In *Econometrics of planning and efficiency*, pages 123–133. Springer, 1988.

[6] Hila Chefer, Shir Gur, and Lior Wolf. Generic attention-model explainability for interpreting bi-modal and encoder-decoder transformers. *arXiv preprint arXiv:2103.15679*, 2021.

[7] Hila Chefer, Shir Gur, and Lior Wolf. Transformer interpretability beyond attention visualization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 782–791, 2021.

[8] Qiwei Chen, Huan Zhao, Wei Li, Pipei Huang, and Wenwu Ou. Behavior sequence transformer for e-commerce recommendation in alibaba. In *Proceedings of the 1st International Workshop on Deep Learning Practice for High-Dimensional Sparse Data*, pages 1–4, 2019.

[9] Edward Choi, Mohammad Taha Bahadori, Joshua A Kulas, Andy Schuetz, Walter F Stewart, and Jimeng Sun. Retain: An interpretable predictive model for healthcare using reverse time attention mechanism. *arXiv preprint arXiv:1608.05745*, 2016.

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

[11] Maurizio Ferrari Dacrema, Paolo Cremonesi, and Dietmar Jannach. Are we really making much progress? a worrying analysis of recent neural recommendation approaches. In *Proceedings of the 13th ACM Conference on Recommender Systems*, pages 101–109, 2019.

[12] Maria De-Arteaga, Alexey Romanov, Hanna Wallach, Jennifer Chayes, Christian Borgs, Alexandra Chouldechova, Sahin Geyik, Krishnaram Kenthapadi, and Adam Tauman Kalai. Bias in bios: A case study of semantic representation bias in a high-stakes setting. In *proceedings of the Conference on Fairness, Accountability, and Transparency*, pages 120–128, 2019.

[13] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[14] Andrea Galassi, Marco Lippi, and Paolo Torroni. Attention in natural language processing. *IEEE Transactions on Neural Networks and Learning Systems*, 2020.

[15] F Maxwell Harper and Joseph A Konstan. The movielens datasets: History and context. *Acm transactions on interactive intelligent systems (tiis)*, 5(4):1–19, 2015.

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[17] Ruining He, Wang-Cheng Kang, and Julian McAuley. Translation-based recommendation. In *Proceedings of the eleventh ACM conference on recommender systems*, pages 161–169, 2017.

[18] Ruining He and Julian McAuley. Fusing similarity models with markov chains for sparse sequential recommendation. In *2016 IEEE 16th International Conference on Data Mining (ICDM)*, pages 191–200. IEEE, 2016.

[19] Ruining He and Julian McAuley. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *proceedings of the 25th international conference on world wide web*, pages 507–517, 2016.

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

[21] 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*, pages 241–250, 2000.

[22] Balázs Hidasi and Alexandros Karatzoglou. Recurrent neural networks with top-k gains for session-based recommendations. In *Proceedings of the 27th ACM international conference on information and knowledge management*, pages 843–852, 2018.

[23] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. Session-based recommendations with recurrent neural networks. *arXiv preprint arXiv:1511.06939*, 2015.

[24] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. Session-based recommendations with recurrent neural networks. *arXiv preprint arXiv:1511.06939*, 2015.

[25] Sarthak Jain and Byron C Wallace. Attention is not explanation. *arXiv preprint arXiv:1902.10186*, 2019.

[26] Wang-Cheng Kang and Julian McAuley. Self-attentive sequential recommendation. In *2018 IEEE International Conference on Data Mining (ICDM)*, pages 197–206. IEEE, 2018.

[27] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[28] Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, 2009.

[29] Vivian Lai and Chenhao Tan. On human predictions with explanations and predictions of machine learning models: A case study on deception detection. In *Proceedings of the conference on fairness, accountability, and transparency*, pages 29–38, 2019.

[30] Gottfried Wilhelm Leibniz. Nova methodus pro maximis et minimis. *Acta Eruditorum*, 1684.

[31] Jiwei Li, Will Monroe, and Dan Jurafsky. Understanding neural networks through representation erasure. *arXiv preprint arXiv:1612.08220*, 2016.

[32] Zachary C Lipton. The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery. *Queue*, 16(3):31–57, 2018.

[33] Scott Lundberg and Su-In Lee. A unified approach to interpreting model predictions. *arXiv preprint arXiv:1705.07874*, 2017.

[34] Andre Martins and Ramon Astudillo. From softmax to sparsemax: A sparse model of attention and multi-label classification. In *International conference on machine learning*, pages 1614–1623. PMLR, 2016.

[35] Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton Van Den Hengel. Image-based recommendations on styles and substitutes. In *Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval*, pages 43–52, 2015.

[36] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[37] Christoph Molnar. *Interpretable machine learning*. Lulu. com, 2020.

[38] Isaac Newton. Philosophiæ naturalis principia mathematica. 1687.

[39] Danish Pruthi, Mansi Gupta, Bhuwan Dhingra, Graham Neubig, and Zachary C Lipton. Learning to deceive with attention-based explanations. *arXiv preprint arXiv:1909.07913*, 2019.

[40] Danish Pruthi, Mansi Gupta, Bhuwan Dhingra, Graham Neubig, and Zachary C Lipton. Learning to deceive with attention-based explanations. *arXiv preprint arXiv:1909.07913*, 2019.

[41] Steffen Rendle. Factorization machines. In *2010 IEEE International Conference on Data Mining*, pages 995–1000. IEEE, 2010.

[42] 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*, pages 1135–1144, 2016.

[43] Francesco Ricci, Lior Rokach, and Bracha Shapira. Recommender systems handbook. Springer, 2011.

[44] Cynthia Rudin and Berk Ustun. Optimized scoring systems: Toward trust in machine learning for healthcare and criminal justice. *Interfaces*, 48(5):449–466, 2018.

[45] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017.

[46] Sofia Serrano and Noah A Smith. Is attention interpretable? *arXiv preprint arXiv:1906.03731*, 2019.

[47] Guy Shani and Asela Gunawardana. Evaluating recommendation systems. In *Recommender systems handbook*, pages 257–297. Springer, 2011.

[48] Lloyd S Shapley. A value for n-person games. *Contributions to the Theory of Games*, 2(28):307–317, 1953.

[49] Avanti Shrikumar, Peyton Greenside, Anna Shcherbina, and Anshul Kundaje. Not just a black box: Learning important features through propagating activation differences. *arXiv preprint arXiv:1605.01713*, 2016.

[50] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013.

[51] Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda Viégas, and Martin Wattenberg. Smoothgrad: removing noise by adding noise. *arXiv preprint arXiv:1706.03825*, 2017.

[52] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

[53] Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. Bert4rec: Sequential recommendation with bidirectional encoder representations from transformer. In *Proceedings of the 28th ACM international conference on information and knowledge management*, pages 1441–1450, 2019.

[54] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *International Conference on Machine Learning*, pages 3319–3328. PMLR, 2017.

[55] Jiaxi Tang and Ke Wang. Personalized top-n sequential recommendation via convolutional sequence embedding. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, pages 565–573, 2018.

[56] Wilson L Taylor. “cloze procedure”: A new tool for measuring readability. *Journalism quarterly*, 30(4):415–433, 1953.

[57] Nava Tintarev and Judith Masthoff. Effective explanations of recommendations: user-centered design. In *Proceedings of the 2007 ACM conference on Recommender systems*, pages 153–156, 2007.

[58] Nava Tintarev and Judith Masthoff. Designing and evaluating explanations for recommender systems. In *Recommender systems handbook*, pages 479–510. Springer, 2011.

[59] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[60] Sarah Wiegreffe and Yuval Pinter. Attention is not not explanation. *arXiv preprint arXiv:1908.04626*, 2019.

[61] Qizhe Xie, Xuezhe Ma, Zihang Dai, and Eduard Hovy. An interpretable knowledge transfer model for knowledge base completion. *arXiv preprint arXiv:1704.05908*, 2017.

[62] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*, pages 2048–2057. PMLR, 2015.

[63] Yongfeng Zhang and Xu Chen. Explainable recommendation: A survey and new perspectives. *arXiv preprint arXiv:1804.11192*, 2018.

[64] Yongfeng Zhang, Guokun Lai, Min Zhang, Yi Zhang, Yiqun Liu, and Shaoping Ma. 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*, pages 83–92, 2014.