

# Faithfully Explaining Rankings in a News Recommender System

Maartje ter Hoeve  
University of Amsterdam\*

Anne Schuth  
De Persgroep

Daan Odijk  
RTL

Maarten de Rijke  
University of Amsterdam

## Summary

- There is an increasing demand for algorithms to be **explainable**
- Explainability of **ranking algorithms** has never been fully addressed
- We propose:
  - **LISTEN**, a faithful **LIST**wise **ExplaiNer**
  - **Q-LISTEN**, a **Quick** version of LISTEN that can run in **real time**
- We show that:
  - LISTEN produces **faithful** explanations
  - Q-LISTEN can **learn** the explanation space created by LISTEN
  - (Q-)LISTEN is **safe to run in production**

## Problem setting

- Blendle: a **news recommender**
- Blendle users receive a **personalised selection** of news articles from a wide variety of news papers every single day
- Item scores for each calculated based on **user-item features**
- Blendle already had **heuristic reasons** as explanations, which we use as **baseline**

### Data

- Historical feature data of Blendle users - split in train, validate and test

## Method: (Q-)LISTEN

- **Intuition:** if changing a feature value heavily affects the ranking, this feature was important for the **item's position** in the ranking, if not, it was not

### LISTEN - Overview

#### Training phase

- 1) Find the **importance** of individual feature values by **changing** them and see how these changes affect the ranking
- 2) Find **points of interest**

#### Explanation phase

- 3) **Use the points of interest** to find the most important features by observing which changes in feature values affect the ranking most
- 4) **Return** the most important features
- 5) The most important features are the **explanations**. Return these to the users in an **understandable way**

### Q-LISTEN

- LISTEN is not efficient enough to run in production in real time
- We train an MLP to learn the explanation space and call this model Q-LISTEN
- Q-LISTEN can generate explanations **in real time**

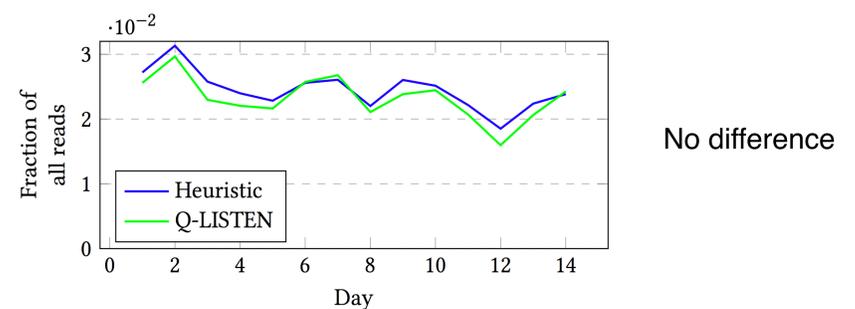
## Results

### RQ1 - Do LISTEN and Q-LISTEN produce faithful explanations of rankings?

- We construct **dummy data** with labels
- LISTEN returns the correct labels, speed up steps slightly decrease the accuracy, yet significantly increase the efficiency
- Our generated explanations are **faithful**

### RQ2 - Does the type of explanation affect users' behaviour?

A/B test on two groups of Blendle users: heuristic vs Q-LISTEN



## Conclusion

- We have investigated the **explainability** of **ranking algorithms**
- We introduced **LISTEN** and **Q-LISTEN**
- LISTEN finds the **most important features** for an item's position in the ranking
- Q-LISTEN allows us to generate explanations for items in the ranking **in production in real time**
- An A/B-test with reasons produced by different explanations systems showed that the **reading behaviour of users does not differ** depending on the type of explanations they see
- It is **safe and preferred to use (Q-)LISTEN in production**, as (Q-)LISTEN outperforms the baseline from a transparency point of view