UNIVERSITEIT VAN AMSTERDAM

# Exploring Architectures and Prompts for Sequential Generative News Recommendation with mT5

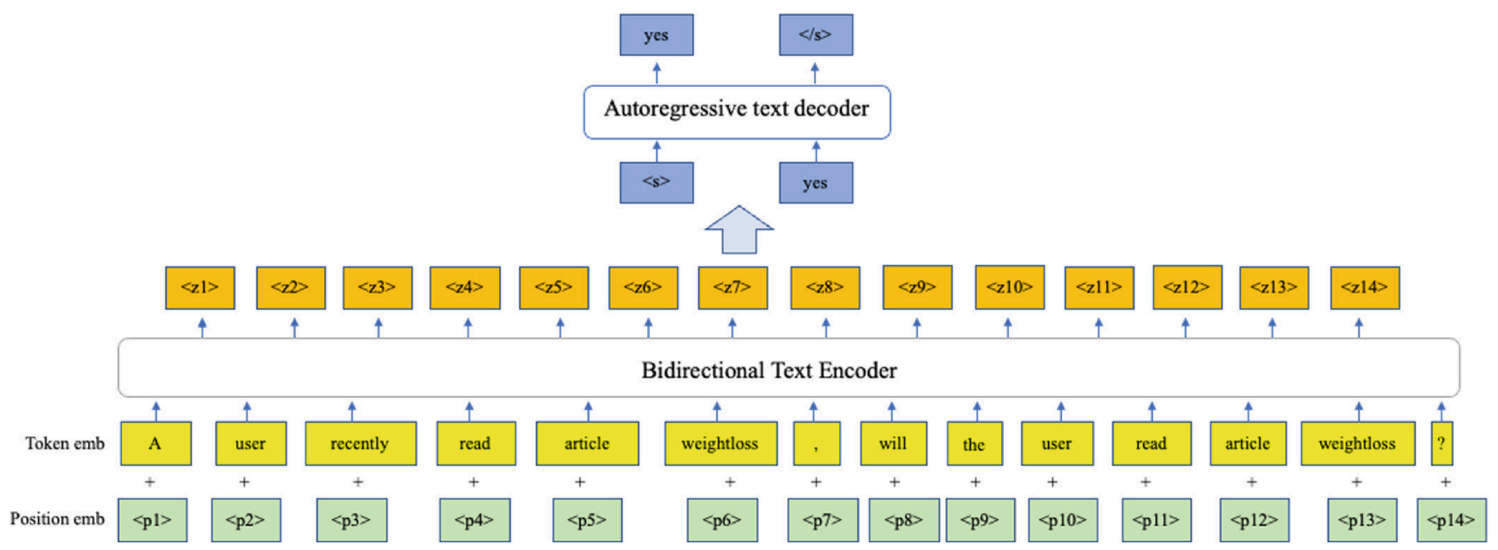
RecSys Challenge 2024
Sponsored by
JP|PolitikensHus Ekstra Bladet

*Wouter Bant, Colin Bot, Maarten Drinhuyzen*
*Supervised by: Yougang Lyu*

## 1. Introduction

### LLM for News Recommendation Systems

PGNR by Li et al (2024)




$$L = (1-\lambda)\,L_{NLL} + \lambda\,L_{BPR}$$

$$L_{NLL} = -\sum_t \log(P_\theta(y_t \mid y_{<t}, X))$$

$$L_{BPR} = -\sum_{(u,pos,neg)} \log(\sigma(r_{u,pos} - r_{u,neg}))$$

**Contributions:**

- Replicate PGNR with new code and different data
- Modify PGNR for efficiency and controllability
- Participate in the RecSys challenge

## 2. Challenges and proposed solutions

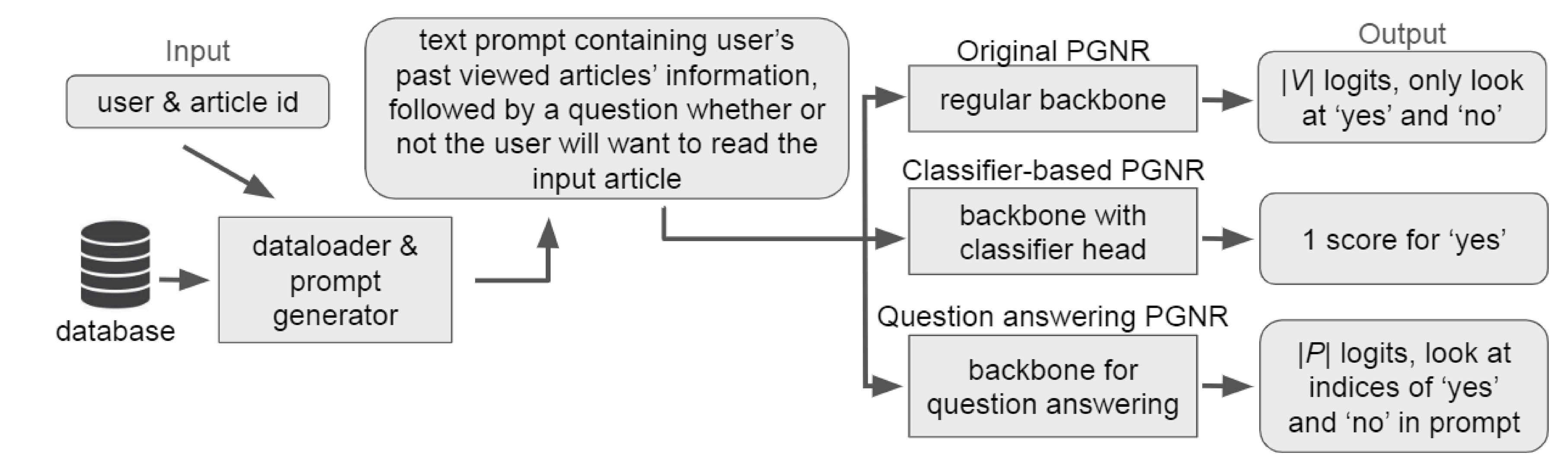
- The provided code doesn't run
  - Write easily extendible code from scratch
- Articles for RecSys challenge are in Danish
  - Use mT5 instead of T5 (350M parameters)
- The proposed architecture (CG) has many redundant parameters, the last layer maps with 128M parameters to 250K logits of which only 2 are used.
  - CGc: maps to 1 number: the score for the article
  - QA: predicts a correct span in a given prompt ("ja / nej")
  - QA+: similar to QA but scores all articles simultaneously

## 3. Further extensions

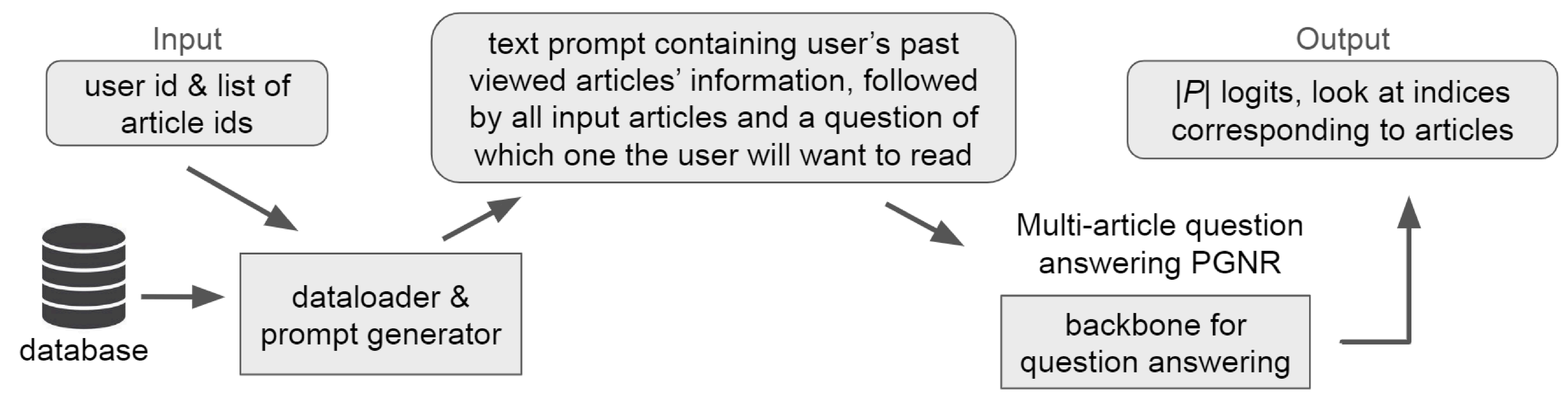
- Investigate encoding more information into the prompt
- Investigate the effect of the rank loss introduced by PGNR
- Investigate the accuracy, efficiency, and controllability of the various models and prompts

## 4. Methodology

### Pipeline CG, CGc, and QA




### Pipeline QA+




### Prompt variations

Prompt templates a, b, c, d (English version of Danish prompts):
*"A user has recently read these articles:* $<article_1>, <article_2>, \cdots, <article_T>$*. Will they read* $<article_{pos}>$*?"*

Prompt template e (English version of Danish prompt):
*"A user has recently read these articles:* $<article_1>, <article_2>, \cdots, <article_T>$*. Which of the following articles will they read* $<in\ view\ article_1>, <in\ view\ article_2>, \cdots, <in\ view\ article_N>$*?"*

$<article_i>$:
a: title
b: subtitle
c: (category) title
d: (time difference between publish time of $i$ and $pos$) title
e: title

$<article_{pos}>$:
a: title
b: subtitle
c: (category) title
d: title

$<in\ view\ article_i>$:
e: title + separator token

## 5. Baselines

CG without fine-tuning:

| Model | Prompt | MRR | HR@1 | HR@5 | NDCG | AUC |
|---|---|---|---|---|---|---|
| CG | a | 0.3108 | 0.1131 | 0.5859 | 0.4667 | 0.5591 |
| | b | **0.3243** | **0.1256** | **0.6003** | **0.4773** | **0.5744** |
| | c | 0.3154 | 0.1182 | 0.5911 | 0.4702 | 0.5644 |
| | d | 0.3111 | 0.1144 | 0.5839 | 0.4669 | 0.5590 |

Random ranking, heuristic methods, and contest winner:

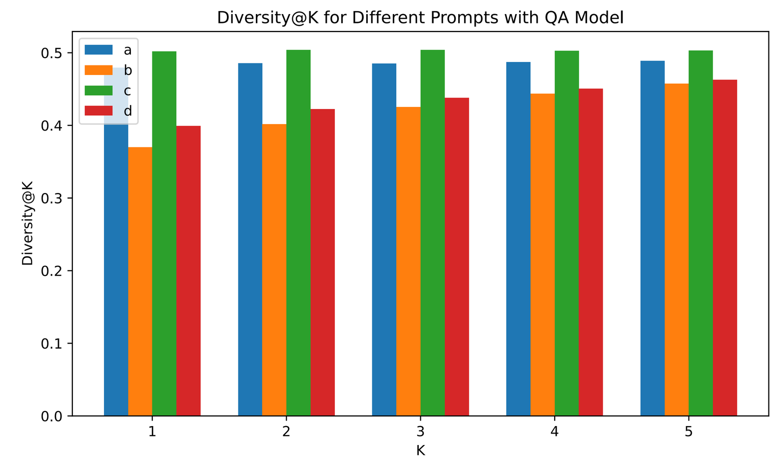
| Model | MRR | HR@1 | HR@5 | NDCG | AUC |
|---|---|---|---|---|---|
| Random | 0.3150 | 0.1185 | 0.2566 | 0.4697 | 0.5604 |
| Most frequent category | 0.3429 | 0.1443 | 0.6205 | 0.4922 | 0.5969 |
| Closest publish time | 0.3896 | **0.1691** | **0.7242** | 0.5322 | 0.6917 |
| Contest winner | **0.7268** | NA | NA | **0.7940** | **0.8767** |

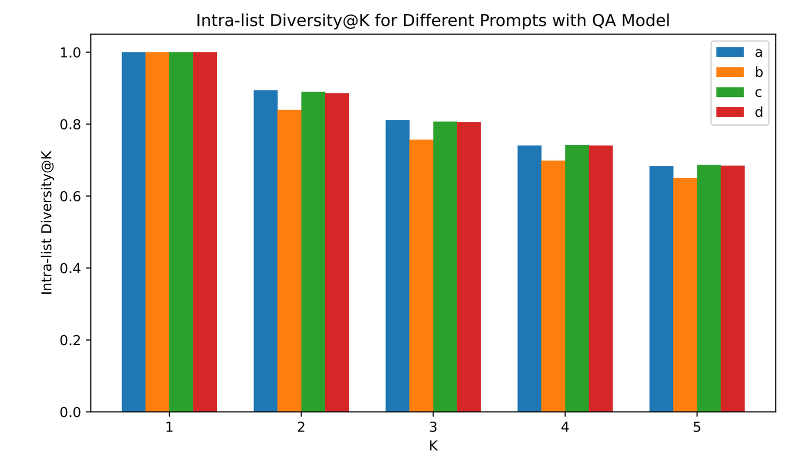
## 6. Results

### Accuracy-based metrics

| | | Training set | | | | | | Validation set | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | $\lambda$=0.0 | | | $\lambda$=0.4 | | | $\lambda$=0.0 | | | $\lambda$=0.4 | | |
| Model | Prompt | MRR | HR@1 | AUC | MRR | HR@1 | AUC | MRR | HR@1 | AUC | MRR | HR@1 | AUC |
| CG | a | 0.4814 | 0.2738 | 0.7428 | 0.4837 | 0.2770 | **0.7448** | 0.3494 | 0.1490 | 0.6101 | 0.3529 | 0.1516 | 0.6167 |
| | b | 0.4786 | 0.2712 | 0.7412 | 0.4777 | 0.2697 | 0.7402 | 0.3629 | 0.1618 | 0.6236 | 0.3622 | 0.1626 | 0.6209 |
| | c | 0.4798 | 0.2721 | 0.7422 | 0.4807 | 0.2735 | 0.7427 | 0.3547 | 0.1525 | 0.6178 | 0.3537 | 0.1523 | 0.6162 |
| | d | 0.4814 | 0.2729 | 0.7428 | 0.4787 | 0.2708 | 0.7408 | 0.3840 | 0.1764 | 0.6568 | 0.3786 | 0.1688 | 0.6551 |
| CGc | a | 0.4830 | 0.2767 | 0.7438 | 0.4824 | 0.2752 | 0.7433 | 0.3483 | 0.1485 | 0.6104 | 0.3476 | 0.1482 | 0.6074 |
| | b | 0.4791 | 0.2713 | 0.7413 | 0.4803 | 0.2729 | 0.7420 | 0.3633 | 0.1647 | 0.6188 | 0.3609 | 0.1606 | 0.6218 |
| | c | 0.4817 | 0.2745 | 0.7433 | **0.4839** | **0.2776** | 0.7442 | 0.3561 | 0.1520 | 0.6222 | 0.3579 | 0.1568 | 0.6203 |
| | d | 0.4816 | 0.2729 | 0.7439 | 0.4825 | 0.2750 | 0.7438 | **0.3864** | **0.1817** | **0.6576** | **0.3834** | **0.1753** | **0.6579** |
| QA | a | 0.4742 | 0.2660 | 0.7373 | 0.4616 | 0.2529 | 0.7261 | 0.3482 | 0.1449 | 0.6132 | 0.3420 | 0.1378 | 0.6096 |
| | b | 0.3224 | 0.1224 | 0.5655 | 0.4747 | 0.2659 | 0.7384 | 0.3153 | 0.1184 | 0.5631 | 0.3612 | 0.1622 | 0.6210 |
| | c | 0.3246 | 0.1244 | 0.5688 | 0.3188 | 0.1209 | 0.5586 | 0.3149 | 0.1204 | 0.5598 | 0.3142 | 0.1187 | 0.5619 |
| | d | 0.4787 | 0.2708 | 0.7408 | 0.4782 | 0.2712 | 0.7404 | 0.3747 | 0.1659 | 0.6503 | 0.3753 | 0.1681 | 0.6520 |
| QA+ | e | **0.5470** | **0.3380** | **0.7940** | 0.4797 | 0.2680 | 0.7436 | 0.3414 | 0.1388 | 0.6054 | 0.3390 | 0.1362 | 0.6028 |

### Beyond-accuracy metrics

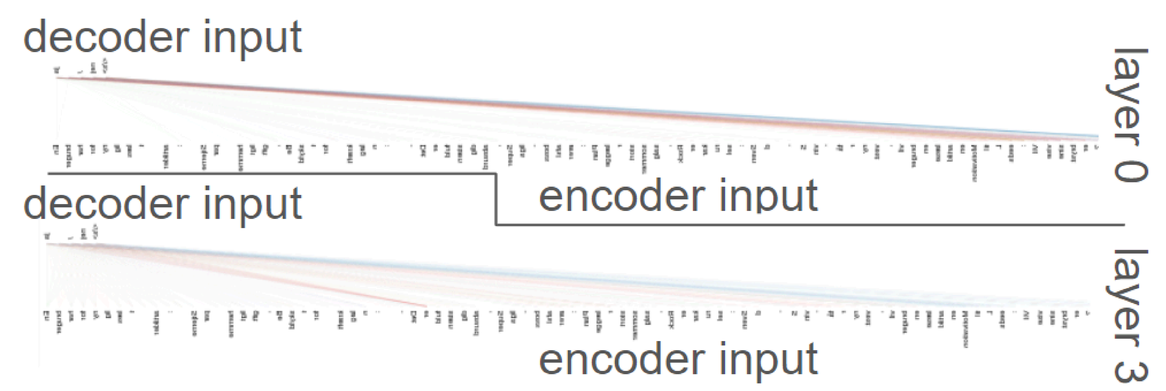



### Model behaviour

Output logits

| input | logit 'ja' | logit 'nej' | avg other logit |
|---|---|---|---|
| positive example | 4.492 ± 0.518 | **5.448** ± 0.456 | -22.783 ± 0.169 |
| negative example | 4.358 ± 0.384 | **5.593** ± 0.399 | -22.825 ± 0.076 |




Cross Attention

## 7. Conclusion

- Poor generalizability to unseen titles/subtitles, however, more information in the prompt can improve this
- CGc is the most accurate and efficient model
- The difference in publishing time makes the best prompt
- Rank loss can help convergence but improvement is minimal
- Different prompts can significantly impact diversity of ranking