FUM: Fine-grained and Fast User Modeling for News Recommendation （SIGIR_2022）

Tao Qi
Department of Electronic Engineering,
Tsinghua University
taoqi.qt@gmail.com

Fangzhao Wu*
Microsoft Research Asia
wufangzhao@gmail.com

Chuhan Wu
Department of Electronic Engineering,
Tsinghua University
wuchuhan15@gmail.com

Yongfeng Huang
Department of Electronic Engineering,
Tsinghua University
yfhuang@tsinghua.edu.cn

2022. 5. 15 • ChongQing

Reported by Lele Duan

1.Background

2.Method

3.Experiments

Background

- Encoding user's clicked news into news embeddings independently and then aggregate them into user embedding.

- The word-level interactions across different clicked news from the same user, which detailed clues to infer user interest, are ignored by these contain rich methods.

	Texts of user's clicked news
1	The challenge in the new story for Iron Man.
2	The upcoming movies of Netflix in 2022.
3	The success of Marvel's Avengers.
4	Adele says if 30 doesn't come now it never will.
5	The most popular songs on YouTube in this week.
6	Vinyl and CD sales both went up in 2021, data says.

Figure 1: The news clicked by a randomly selected user. Word-level relatedness across texts of user's clicked news provide detailed clues to understand user interest.

Over view

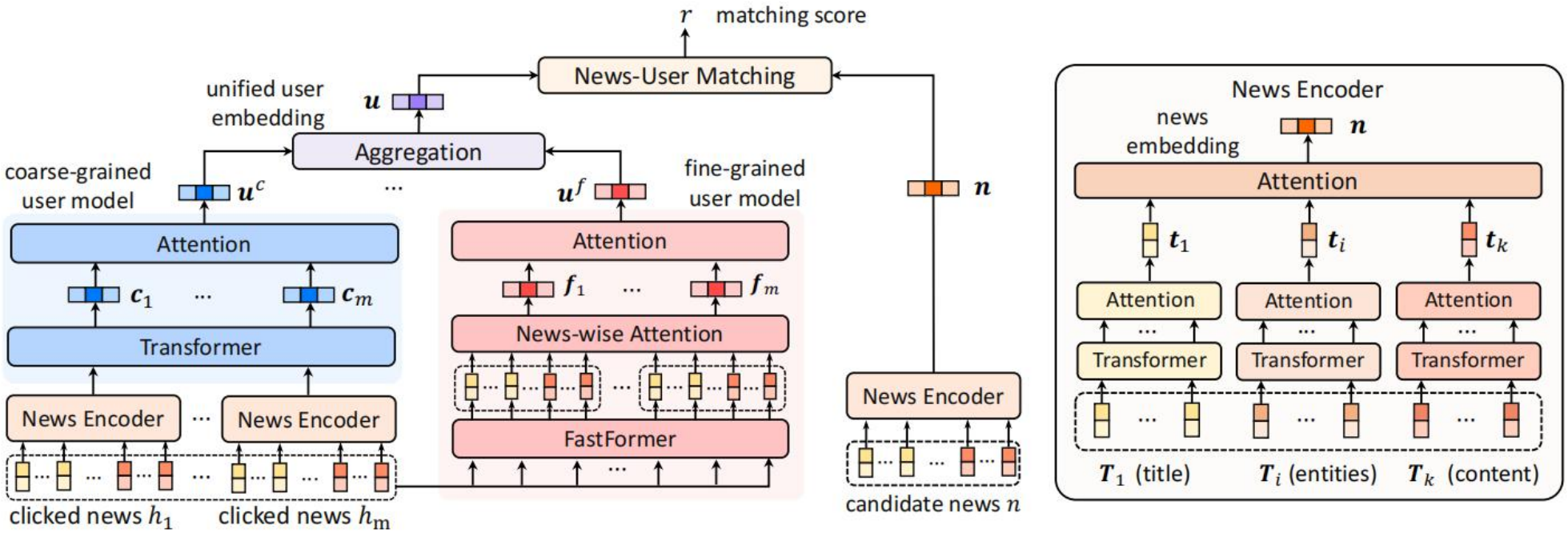


Figure 2: The framework of *FUM* for news recommendation.

Method

- We assume that a news article h contains k genres of textual information (e.g., titles and entities) $[T_1, T_2, ..., T_k]$, where T_i is the i-th genre of the news text.
- $T_i = [t_{i,1}, t_{i,2}, ..., t_{i,l}]$

- We assume that a target user u has previously clicked h_j news, where j-th denotes the j-th clicked news.

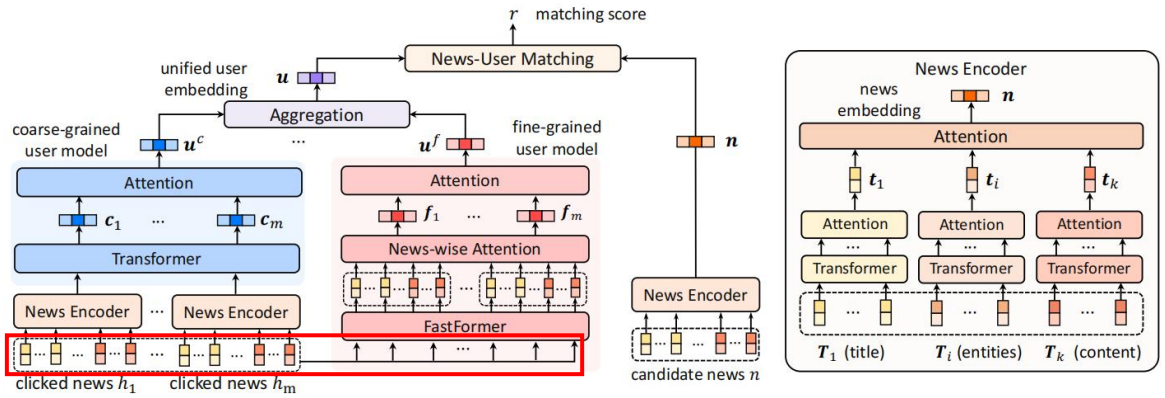


Figure 2: The framework of *FUM* for news recommendation.

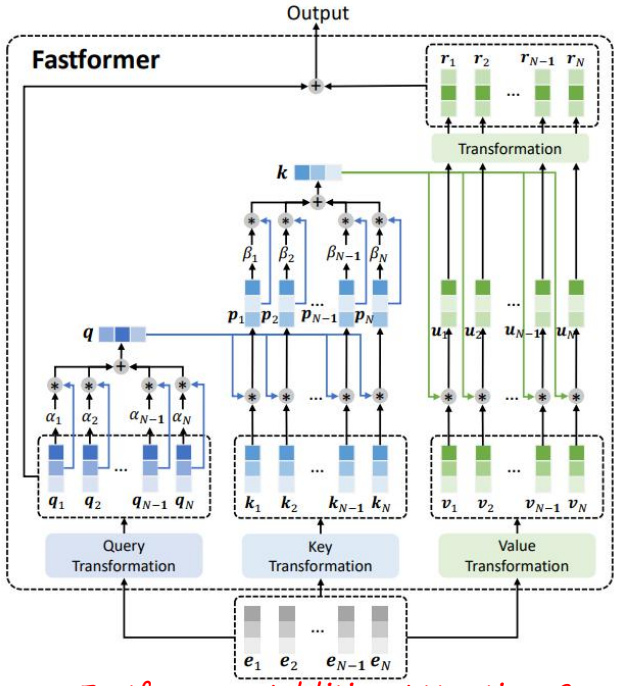
$$\mathbf{T} = [\mathbf{T}_1^1; ...; \mathbf{T}_k^1; ...; \mathbf{T}_1^m; ...; \mathbf{T}_k^m], \quad (1)$$

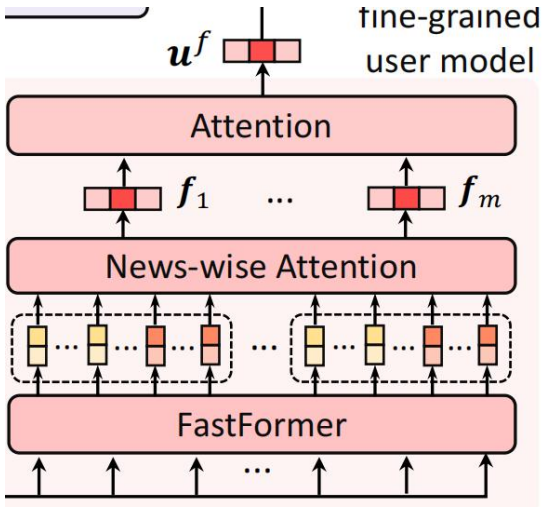
$$\mathbf{T} \in \mathbb{R}^{mkl \times d}$$

$$\mathbf{T}_i \in \mathbb{R}^{l \times d}$$

where $\mathbf{T}_j^i$ is the j-th text embedding sequence of the i-th clicked news h_i and ; is the concatenation operation. Besides, different gen-

In experiments, we utilize news topic labels, description texts of entities, titles, and abstracts for news modeling.





$$\mathbf{T} \in \mathbb{R}^{mkl \times d} \longrightarrow \mathbf{H} \in \mathbb{R}^{L \times g}$$

2021_Fastformer: Additive Attention Can Be All You Need

Figure 1: The architecture of *Fastformer*.

$$\mathbf{q} = Att(\mathbf{q}_1, ..., \mathbf{q}_L), \quad \mathbf{q}_i = \mathbf{W}_q \mathbf{h}_i, \tag{2}$$

$$\mathbf{k} = Att(\mathbf{q} * \mathbf{k}_1, ..., \mathbf{q} * \mathbf{k}_L), \quad \mathbf{k}_i = \mathbf{W}_k \mathbf{h}_i \tag{3}$$

$$\hat{\mathbf{h}}_i = \mathbf{W}_o(\mathbf{k} * \mathbf{v}_i), \quad \mathbf{v}_i = \mathbf{W}_v \mathbf{h}_i \tag{4}$$

where $\mathbf{h}_i$ and $\hat{\mathbf{h}}_i$ denote the input and output of the i-th token in the behavior embedding sequence, $*$ denotes element-wise product, $Att(\cdot)$ denotes the attention pooling network and $\mathbf{W}_q$, $\mathbf{W}_k$, $\mathbf{W}_v$ and $\mathbf{W}_o$ denote trainable projection parameters. We remark that

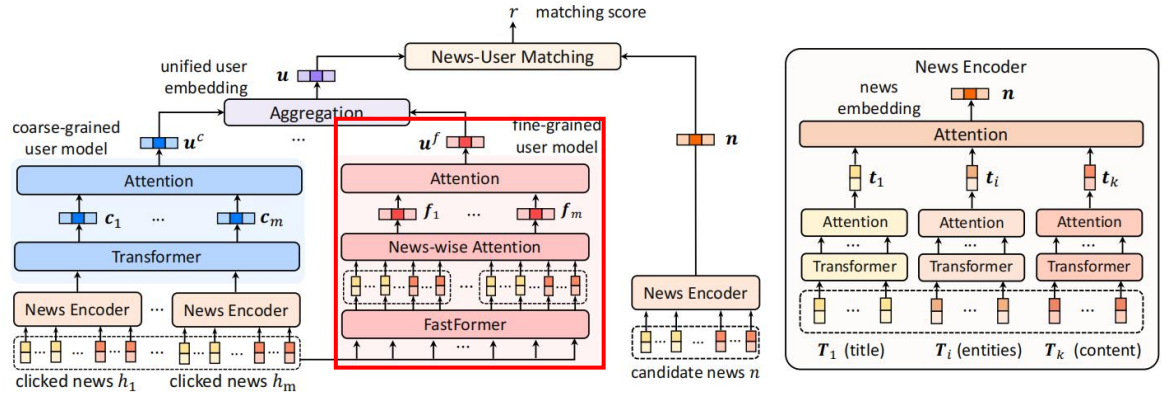


Figure 2: The framework of *FUM* for news recommendation.

$$\mathbf{g}_i = \oplus_{\text{Head}_i} \hat{\mathbf{h}}_i$$
$$\mathbf{f}_i = Att(\mathbf{g}_{(i-1)kl+1}, \mathbf{g}_{(i-1)kl+2}, ..., \mathbf{g}_{ikl}), \tag{5}$$

where $\mathbf{f}_i$ represents the i-th clicked news. Finally, we pooling them to build the user embedding $\mathbf{u}^f = Att(\mathbf{f}_1, ..., \mathbf{f}_m)$. In this way, we can efficiently and effectively model and encode user interest from word-level fine-grained behavior interactions.

$$\mathbf{u}^c = Att(\mathbf{c}_1, ..., \mathbf{c}_m)$$
$$\mathbf{u} = \mathbf{u}^f + \mathbf{u}^c.$$
$$r = \mathbf{u}^T \mathbf{n}.$$

$$\mathcal{L} = -\frac{1}{|\mathcal{D}|} \sum_{i=1}^{|\mathcal{D}|} \sigma(r_i^c - r_i^n)$$

Table 1: News recommendation performance of different methods on *MIND* and *Feeds*. The improvement of *FUM* over baseline methods is significant at level $p < 0.001$ based on t-test.

	MIND				*Feeds*			
	AUC	MRR	nDCG@5	nDCG@10	AUC	MRR	nDCG@5	nDCG@10
GRU	65.47±0.18	31.15±0.22	33.64±0.24	39.34±0.24	62.95±0.13	27.57±0.08	31.55±0.12	37.18±0.11
DKN	67.19±0.13	32.97±0.19	35.87±0.22	41.53±0.17	64.02±0.25	28.65±0.13	32.97±0.17	38.54±0.17
NPA	67.42±0.15	32.97±0.18	35.90±0.23	41.54±0.20	64.83±0.47	29.21±0.36	33.64±0.47	39.18±0.48
KRED	67.77±0.15	33.39±0.15	36.34±0.17	42.04±0.15	64.92±0.14	29.27±0.08	33.71±0.13	39.25±0.12
GNewsRec	68.38±0.09	33.46±0.22	36.44±0.23	42.15±0.20	65.02±0.11	29.28±0.10	33.74±0.13	39.28±0.13
NAML	68.16±0.11	33.31±0.07	36.26±0.10	41.94±0.08	65.31±0.12	29.47±0.07	33.99±0.09	39.57±0.12
NRMS	68.33±0.27	33.55±0.27	36.53±0.32	42.18±0.30	65.21±0.12	29.39±0.05	33.87±0.06	39.46±0.08
LSTUR	68.53±0.10	33.58±0.15	36.54±0.18	42.23±0.17	65.31±0.20	29.54±0.15	34.08±0.19	39.63±0.19
FIM	68.15±0.33	33.36±0.27	36.38±0.30	42.02±0.31	65.47±0.12	29.62±0.07	34.19±0.09	39.72±0.09
FUM	**70.01**±0.10	**34.51**±0.13	**37.68**±0.14	**43.38**±0.13	**66.93**±0.19	**30.49**±0.16	**35.31**±0.21	**40.87**±0.18

Table 2: Efficiency comparison of user modeling methods on both model training and inference based on 1k samples.

	GRU	DKN	NAML	NPA	KRED	GNewsRec	LSTUR	NRMS	FIM	FUM
Training Time	11.46s	8.19s	7.98s	8.10s	10.40s	10.72s	11.53s	11.39s	15.85s	13.21s
Inference Time	2.41s	44.90s	1.23s	1.15s	1.24s	86.90s	2.43s	2.16s	350.38s	2.75s
Cacheable	✓	✗	✓	✓	✓	✗	✓	✓	✗	✓

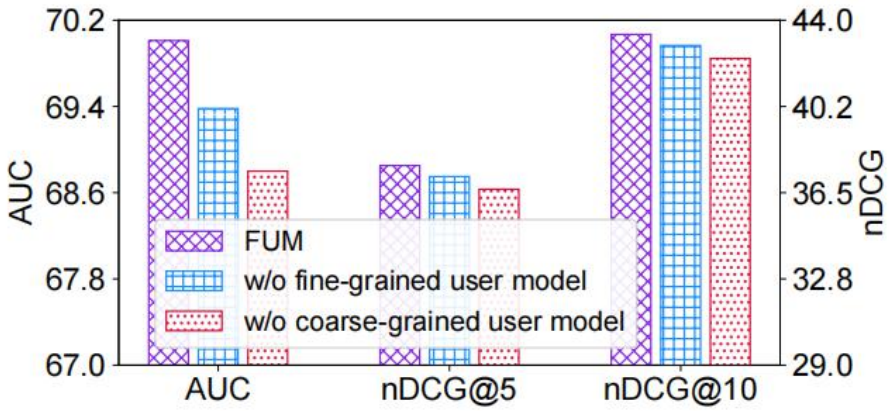


Figure 3: Ablation study of our *FUM* approach.

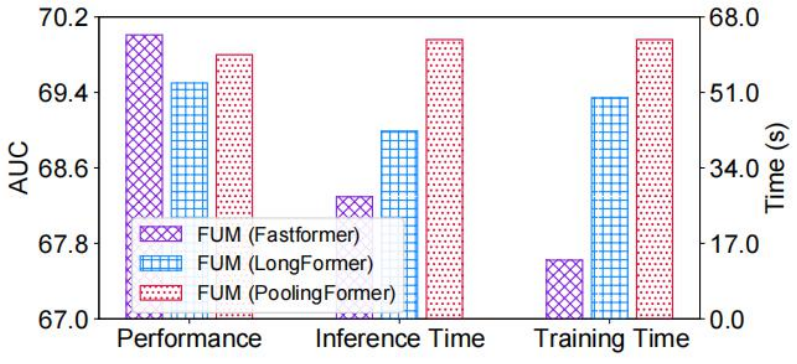


Figure 4: *FUM* with different efficient transformers. The training and inference time are based on 1k and 10k samples.

Thank you!