DKN: Deep Knowledge-Aware Network for News Recommendation

Hongwei Wang[1,2], Fuzheng Zhang[2], Xing Xie[2], Minyi Guo[1*]
[1]Shanghai Jiao Tong University, Shanghai, China
[2]Microsoft Research Asia, Beijing, China
wanghongwei55@gmail.com,{fuzzhang,xingx}@microsoft.com,guo-my@cs.sjtu.edu.cn

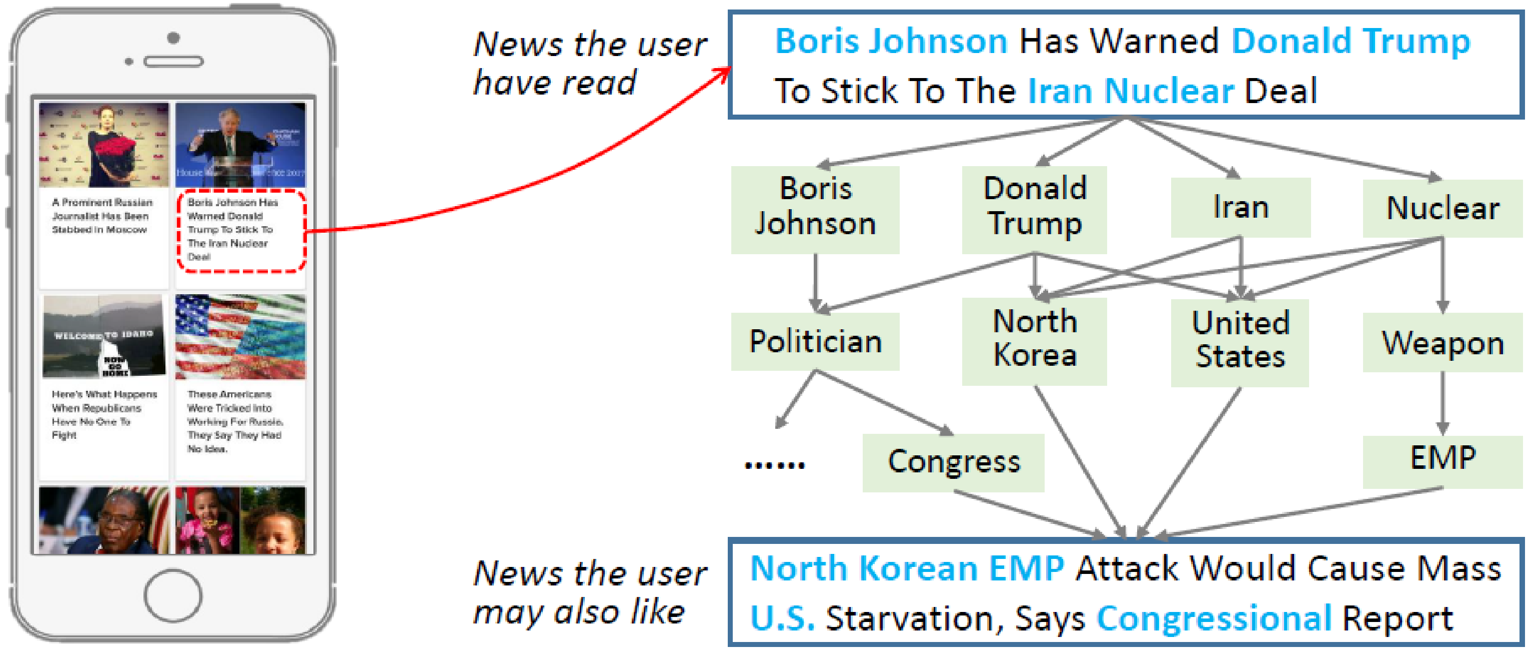
霍沛

Content

- Intention and background
- Basic knowledge introduction
- Model architecture
- Experiments and results
- Conclusion

Intention and background

Goal： Explosion of news and make personalized recommendation



Major challenges:

✓ Highly time-sensitive

✓ Topic-sensitive and Multi-interested user

✓ Highly condensed and comprised of a large amount of knowledge entities and common sense.

Basic knowledge introduction

Knowledge Graph Embedding

- 表示图中的实体和关系
- 定义一个打分函数(scoring function)
- 学习实体和关系的向量表示

- TransE

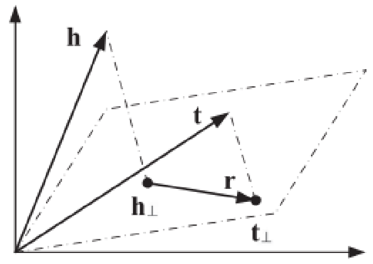
$$\mathbf{h} + \mathbf{r} \approx \mathbf{t}$$

score： $f_r(h,t) = -\|\mathbf{h}_\perp + \mathbf{r} - \mathbf{t}_\perp\|_2^2,$

- TransH   hyperplanes

$$\mathbf{h}_\perp = \mathbf{h} - \mathbf{w}_r^\top \mathbf{h} \mathbf{w}_r, \quad \mathbf{t}_\perp = \mathbf{t} - \mathbf{w}_r^\top \mathbf{t} \mathbf{w}_r.$$

$$\mathbf{h}_\perp + \mathbf{r} \approx \mathbf{t}_\perp$$

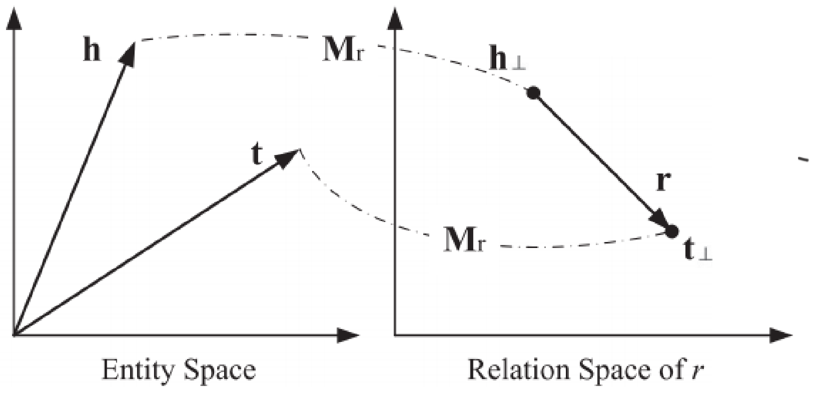


(b) TransH.

- TransR

Entity space $\mathbb{R}^d$, $\mathbf{h}_\perp = \mathbf{M}_r \mathbf{h}, \quad \mathbf{t}_\perp = \mathbf{M}_r \mathbf{t}.$

Each relation $\mathbb{R}^k$ $\mathbf{M}_r \in \mathbb{R}^{k \times d}$



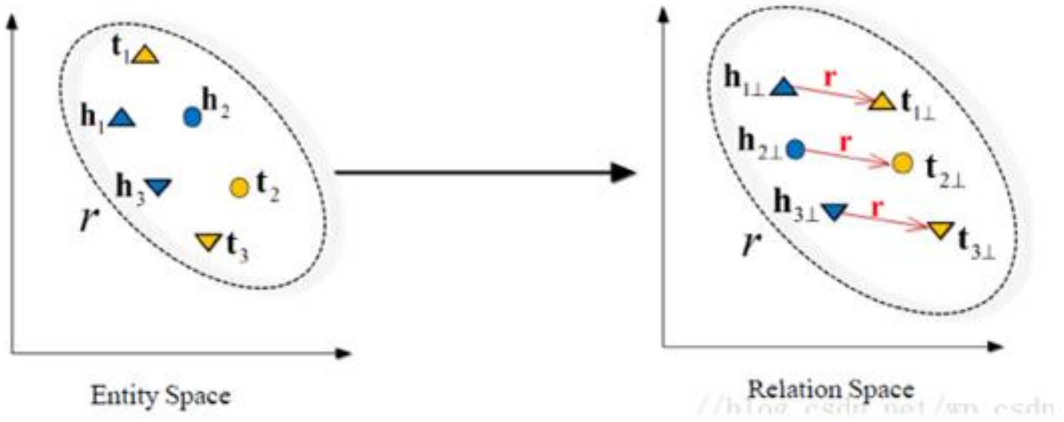
(c) TransR.

- TransD

$$\mathbf{M}_r^1 = \mathbf{w}_r \mathbf{w}_h^\top + \mathbf{I}, \quad \mathbf{M}_r^2 = \mathbf{w}_r \mathbf{w}_t^\top + \mathbf{I}.$$

$$\mathbf{w}_h, \mathbf{w}_t \in \mathbb{R}^d \quad \mathbf{w}_r \in \mathbb{R}^k,$$

$$\mathbf{h}_\perp = \mathbf{M}_r^1 \mathbf{h}, \quad \mathbf{t}_\perp = \mathbf{M}_r^2 \mathbf{t}.$$



损失函数：

$$\mathcal{L} = \sum_{(h,r,t)\in\Delta} \sum_{(h',r,t')\in\Delta'} \max\left(0, f_r(h,t) + \gamma - f_r(h',t')\right),$$

CNN for Sentence Representation Learning

- a sentence of length $\mathbf{w}_{1:n} = [\mathbf{w}_1\ \mathbf{w}_2\ \dots\ \mathbf{w}_n]$ $\mathbb{R}^{d\times n}$
- convolution operation with filter $h \in \mathbb{R}^{d\times l}$

$$c_i = f(h * \mathbf{w}_{i:i+l-1} + b),$$

$$\mathbf{c} = [c_1, c_2, \dots, c_{n-l+1}]$$

- max-over-time pooling operation

$$\tilde{c} = \max\{\mathbf{c}\} = \max\{c_1, c_2, \dots, c_{n-l+1}\}.$$

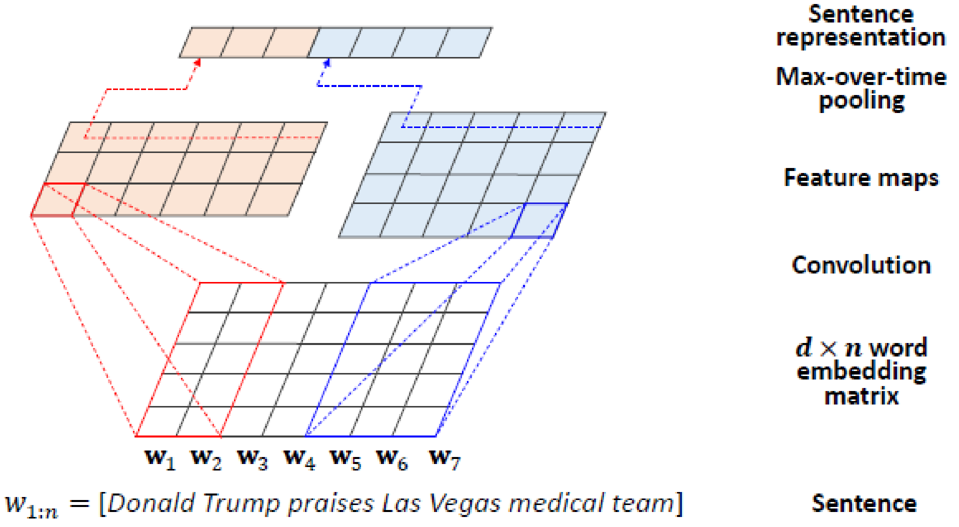


Figure 2: A typical architecture of CNN for sentence representation learning [20].

Problem Formulation

user i

click history: $\{t_1^i, t_2^i, ..., t_{N_i}^i\}$

aim to predict whether user i will click a candidate news tj that he has not seen before

DKN Framework

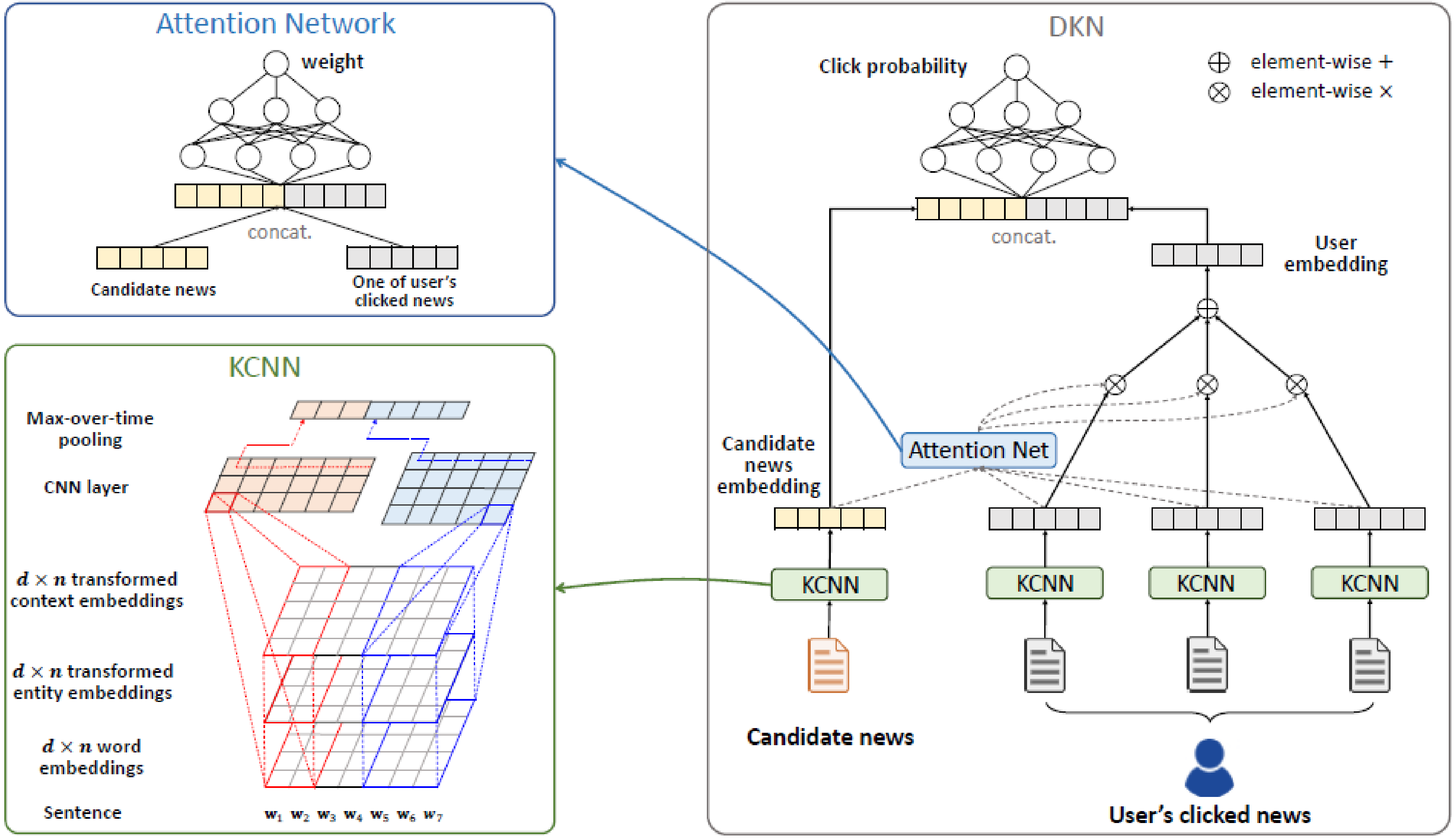


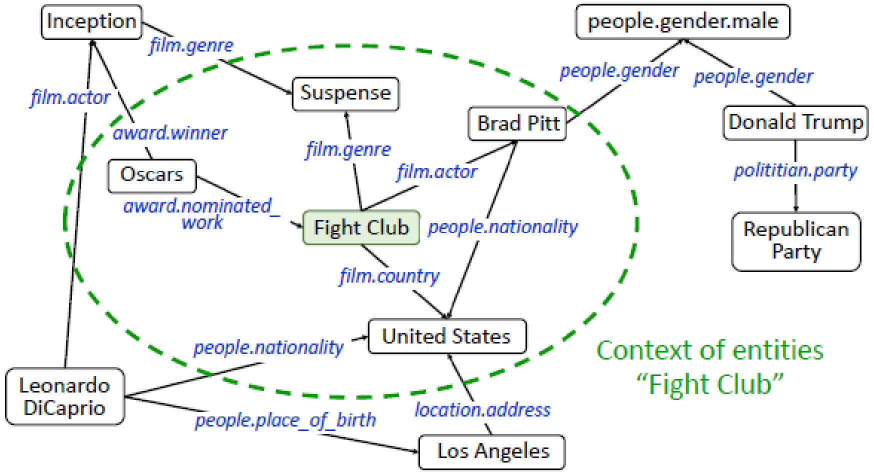
Figure 3: Illustration of the DKN framework.

Knowledge Distillation

- By the technique of entity linking to identify entities
- Construct a sub-graph
- Knowledge graph embedding
- additional contextual information for each entity

$$context(e) = \{e_i \mid (e, r, e_i) \in \mathcal{G} \text{ or } (e_i, r, e) \in \mathcal{G}\}$$

$$\bar{\mathrm{e}} = \frac{1}{|context(e)|} \sum_{e_i \in context(e)} \mathrm{e}_i,$$



Knowledge-aware CNN

- a news title t of length n

$$\mathbf{w}_{1:n} = [\mathbf{w}_1\ \mathbf{w}_2\ \ldots\ \mathbf{w}_n] \in \mathbb{R}^{d\times n}$$

- knowledge distillation

$$\mathbf{e}_i \in \mathbb{R}^{k\times 1} \qquad \overline{\mathbf{e}}_i \in \mathbb{R}^{k\times 1}$$

- Simply method

$$\mathbf{W} = [\mathbf{w}_1\ \mathbf{w}_2\ \ldots\ \mathbf{w}_n\ \mathbf{e}_{t_1}\ \mathbf{e}_{t_2}\ \ldots],$$

fed into CNN

- multi-channel method

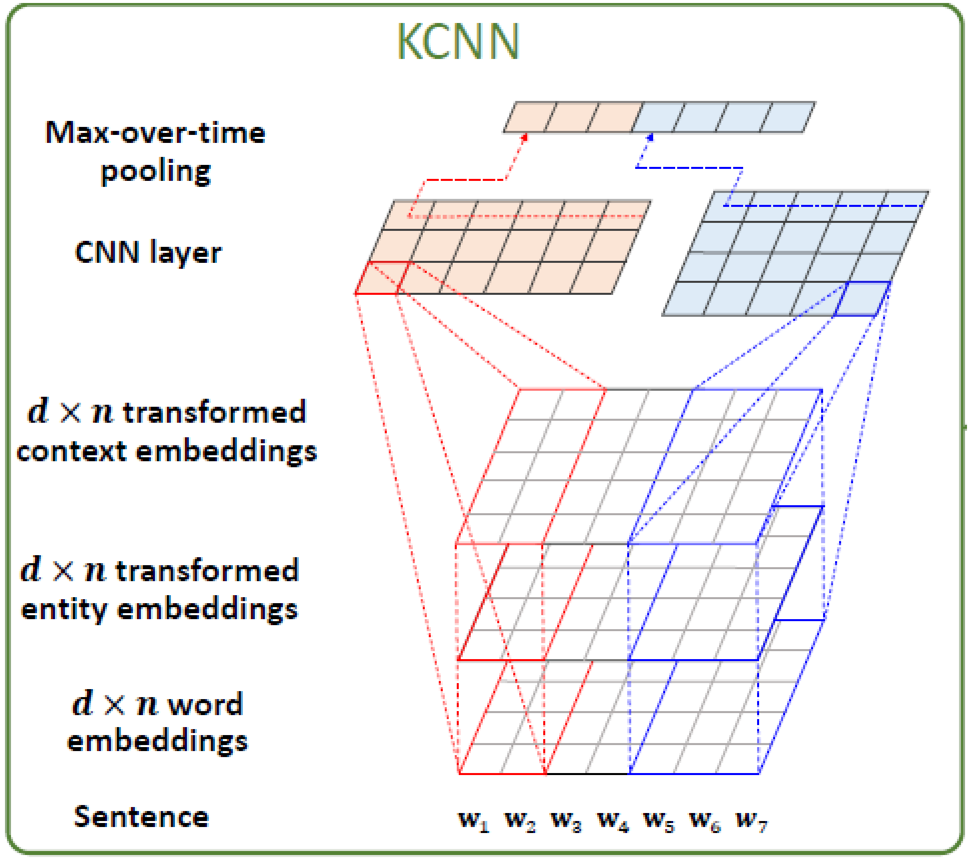
$$g(\mathbf{e}_{1:n}) = [g(\mathbf{e}_1)\ g(\mathbf{e}_2)\ \ldots\ g(\mathbf{e}_n)]$$

$$g(\overline{\mathbf{e}}_{1:n}) = [g(\overline{\mathbf{e}}_1)\ g(\overline{\mathbf{e}}_2)\ \ldots\ g(\overline{\mathbf{e}}_n)]$$

g is the transformation function

$$g(\mathbf{e}) = \mathbf{M}\mathbf{e} \qquad g(\mathbf{e}) = \tanh(\mathbf{M}\mathbf{e} + \mathbf{b}), \qquad \mathbf{M} \in \mathbb{R}^{d\times k}$$

multi-channel input W : $\mathbf{W} = \big[[\mathbf{w}_1\ g(\mathbf{e}_1)\ g(\overline{\mathbf{e}}_1)]\ [\mathbf{w}_2\ g(\mathbf{e}_2)\ \overline{g}(\mathbf{e}_2)]\ \ldots\ [\mathbf{e}_n\ g(\mathbf{e}_n)\ g(\overline{\mathbf{e}}_n)]\big] \in \mathbb{R}^{d\times n\times 3}$

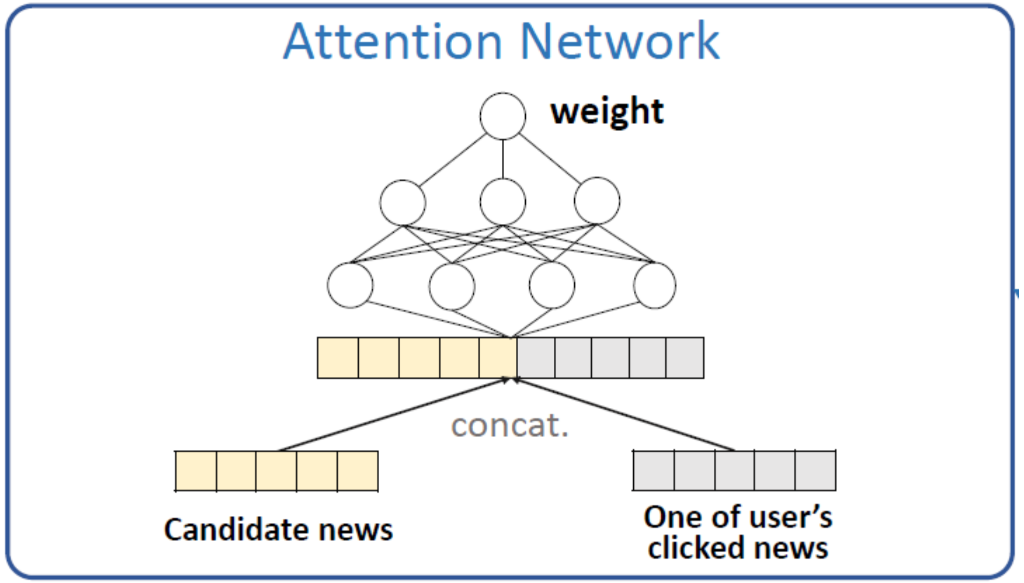


Attention-based User Interest Extraction

user i with clicked history $\{t_1^i, t_2^i, ..., t_{N_i}^i\}$

$\downarrow$ KCNN

$\mathbf{e}(t_1^i), \mathbf{e}(t_2^i), ..., \mathbf{e}(t_{N_i}^i).$



- Simply method

$$\mathbf{e}(i) = \frac{1}{N_i} \sum_{k=1}^{N_i} \mathbf{e}(t_k^i).$$

- Attention network

the candidate news t j   clicked news t_k^i

apply a DNN $\mathcal{H}$

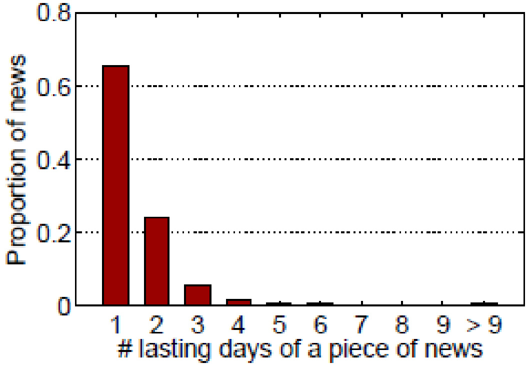
$$s_{t_k^i, t_j} = \text{softmax}\Big(\mathcal{H}\big(\mathbf{e}(t_k^i), \mathbf{e}(t_j)\big)\Big) = \frac{\exp\Big(\mathcal{H}\big(\mathbf{e}(t_k^i), \mathbf{e}(t_j)\big)\Big)}{\sum_{k=1}^{N_i} \exp\Big(\mathcal{H}\big(\mathbf{e}(t_k^i), \mathbf{e}(t_j)\big)\Big)}.$$

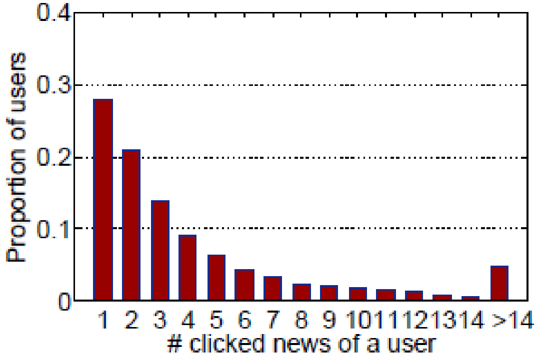
$$\mathbf{e}(i) = \sum_{k=1}^{N_i} s_{t_k^i, t_j} \mathbf{e}(t_k^i)$$

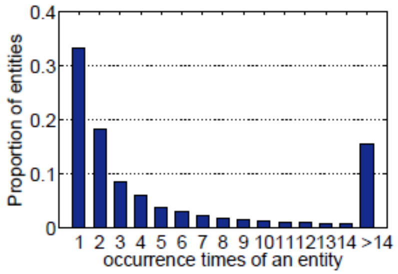
Experiments and results

Dataset:

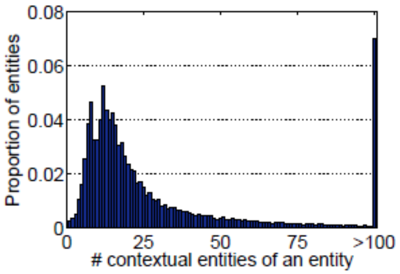
Bing News   Microsoft Satori knowledge graph



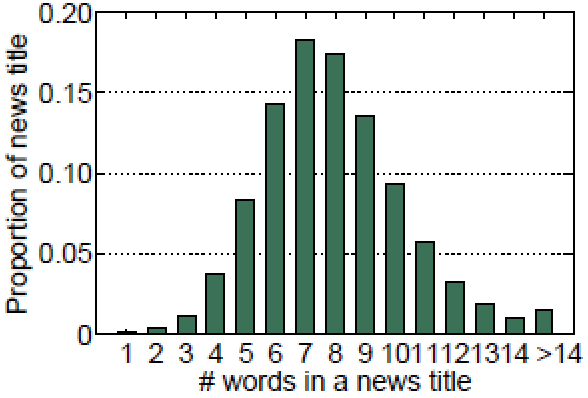


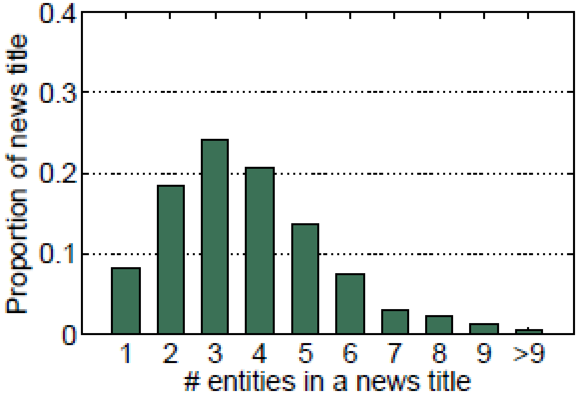


(e) Distribution of the occurrence times of an entity in the news dataset



(f) Distribution of the number of contextual entities of an entity in the knowledge graph





Result:

Table 2: Comparison of different models.

Models*	F1	AUC	p-value**
DKN	**68.9 ± 1.5**	**65.9 ± 1.2**	–
LibFM	61.8 ± 2.1 (-10.3%)	59.7 ± 1.8 (-9.4%)	$< 10^{-3}$
LibFM(-)	61.1 ± 1.9 (-11.3%)	58.9 ± 1.7 (-10.6%)	$< 10^{-3}$
KPCNN	67.0 ± 1.6 (-2.8%)	64.2 ± 1.4 (-2.6%)	0.098
KPCNN(-)	65.8 ± 1.4 (-4.5%)	63.1 ± 1.5 (-4.2%)	0.036
DSSM	66.7 ± 1.8 (-3.2%)	63.6 ± 2.0 (-3.5%)	0.063
DSSM(-)	66.1 ± 1.6 (-4.1%)	63.2 ± 1.8 (-4.1%)	0.045
DeepWide	66.0 ±1.2 (-4.2%)	63.3 ± 1.5 (-3.9%)	0.039
DeepWide(-)	63.7 ± 0.9 (-7.5%)	61.5 ± 1.1 (-6.7%)	0.004
DeepFM	63.8 ± 1.5 (-7.4%)	61.2 ± 2.3 (-7.1%)	0.014
DeepFM(-)	64.0 ± 1.9 (-7.1%)	61.1 ± 1.8 (-7.3%)	0.007
YouTubeNet	65.5 ± 1.2 (-4.9%)	63.0 ± 1.4 (-4.4%)	0.025
YouTubeNet(-)	65.1 ± 0.7 (-5.5%)	62.1 ± 1.3 (-5.8%)	0.011
DMF	57.2 ± 1.2 (-17.0%)	55.3 ± 1.0 (-16.1%)	$< 10^{-3}$

* "(-)" denotes "without input of entity embeddings".

** p-value is the probability of no significant difference with DKN on AUC by t-test.

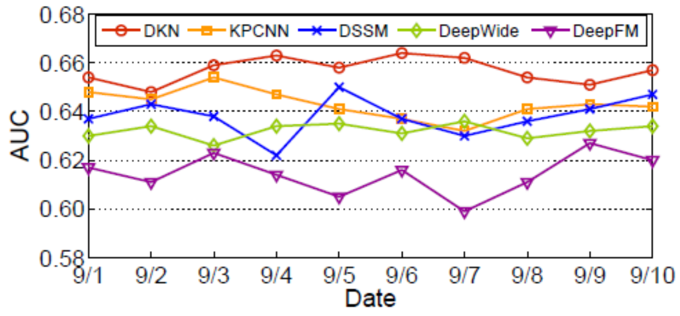


Figure 7: AUC score of DKN and baselines over ten days (Sep. 01-10, 2017).

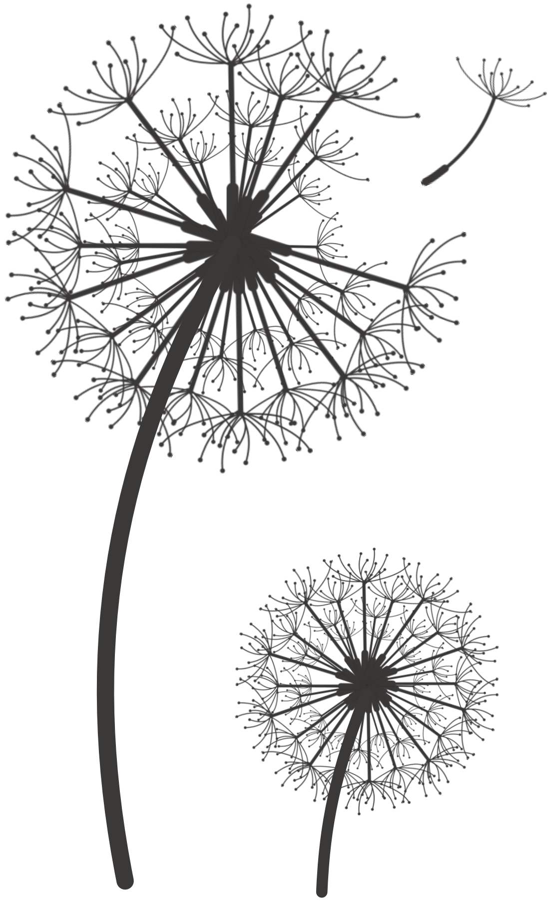
Table 3: Comparison among DKN variants.

Variants	F1	AUC
DKN with entity and context emd.	**68.8 ± 1.4**	**65.7 ± 1.1**
DKN with entity emd. only	67.2 ± 1.2	64.8 ± 1.0
DKN with context emd. only	66.5 ± 1.5	64.2 ± 1.3
DKN without entity nor context emd.	66.1 ±1.4	63.5 ± 1.1
DKN + TransE	67.6 ± 1.6	65.0 ± 1.3
DKN + TransH	67.3 ± 1.3	64.7 ± 1.2
DKN + TransR	67.9 ± 1.5	65.1 ± 1.5
DKN + TransD	**68.8 ± 1.3**	**65.8 ± 1.4**
DKN with non-linear mapping	**69.0 ± 1.7**	**66.1 ± 1.4**
DKN with linear mapping	67.1 ± 1.5	64.9 ± 1.3
DKN without mapping	66.7 ± 1.6	63.7 ± 1.6
DKN with attention	**68.7 ± 1.3**	**65.7 ± 1.2**
DKN without attention	67.0 ± 1.0	64.8 ± 0.8

CONCLUSIONS

对于新闻推荐中存在的问题和特点：新闻具有时效性和较多的实体，有针对性地提出了 DKN 模型，解决了三个挑战：

- DKN 是一个基于内容过滤的深度推荐系统模型；
- 为了利用知识图谱中的信息，通过 KCNN 来融合文本的语义层面、实体层面上的异构表示；
- 使用了注意力机制对用户的兴趣进行动态提取。



THANKS