DAN : Deep Attention Neural Network for News Recommendation

Qiannan Zhu, Xiaofei Zhou, Zengliang Song,Jianlong Tan, Guo Li

[1]Institute of Information Engineering, Chinese Academy of Sciences

[2]University of Chinese Academy of Sciences

AAAI'19 (Oral) 1.27 - 2.1, 2019, Hawaii, USA

Introduction

Observation

- Recommendation System

Problem & Motivation

- ignore the news profile
- ignore the influence of sequential information of a user’s clicked news

Proposal: DAN

- A deep attention neural network DAN that consists of three components including PCNN, ANN, ARNN

Results

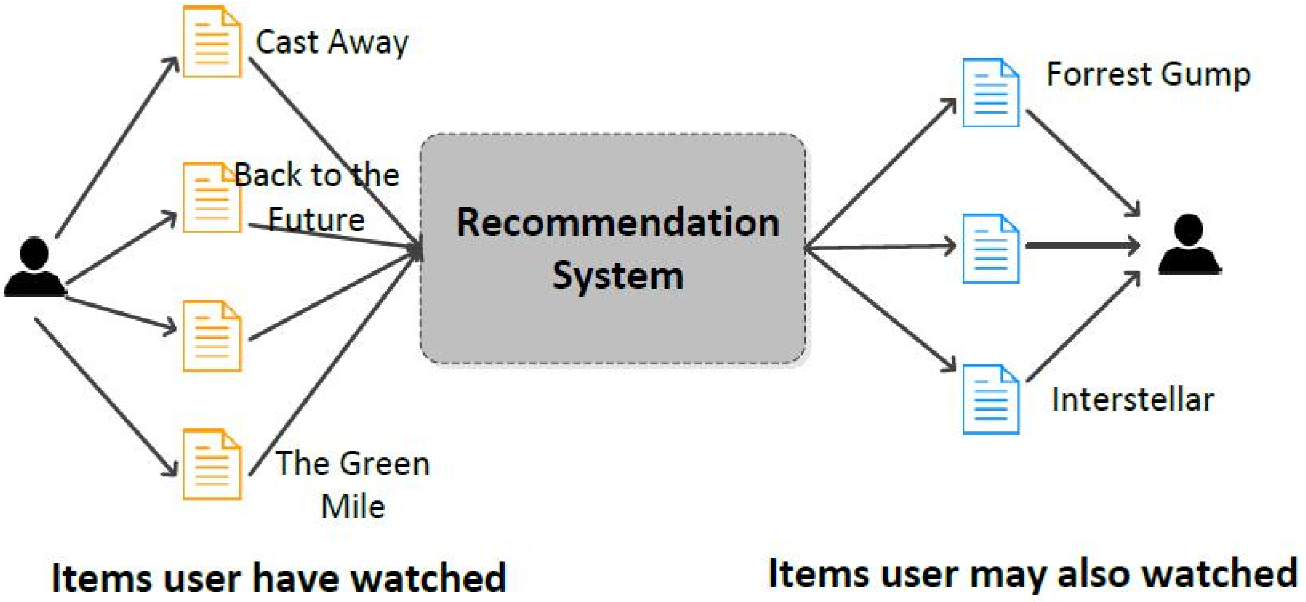
- 3.91% on F1 and 2.64% on AUC improvemnet on evaluation metrices

Contents

Recommendation System

Task

- Given a list of reading history, recommend candidate items for users.



Backgrounds

Recommendation System

Collaborative Filtering (CF) based Methods

- predicts a personalized rankingover a set of items for each individual user with the similarities among the users and items

Content based Methods

- consider the actual content or attributes of the items for making recommendations

Backgrounds

Recommendation System

Hybrid Methods

- recommend items through a hybrid recommender system that usually combines several different recommender algorithms

Deep Learning Based Models

- modeling complex user-item (i.e., news) interactions, and capturing the dynamic properties of news and users

Related Works

Methodologies

- Autoencoders (Sheng,Kawale,and Fu. 2015)
- CDAE (Wu et al. 2016)
- DMF (Xue et al. 2017)
- DSSM (Huang et al. 2013)
- SCENE (Li et al. 2011)
- DeepWide (Cheng et al. 2016),
- DeepFM (Guo et al. 2017),
- DeepJoNN (Zhang, Liu, and Gulla 2018)
- DKN(HongweiWang2018)

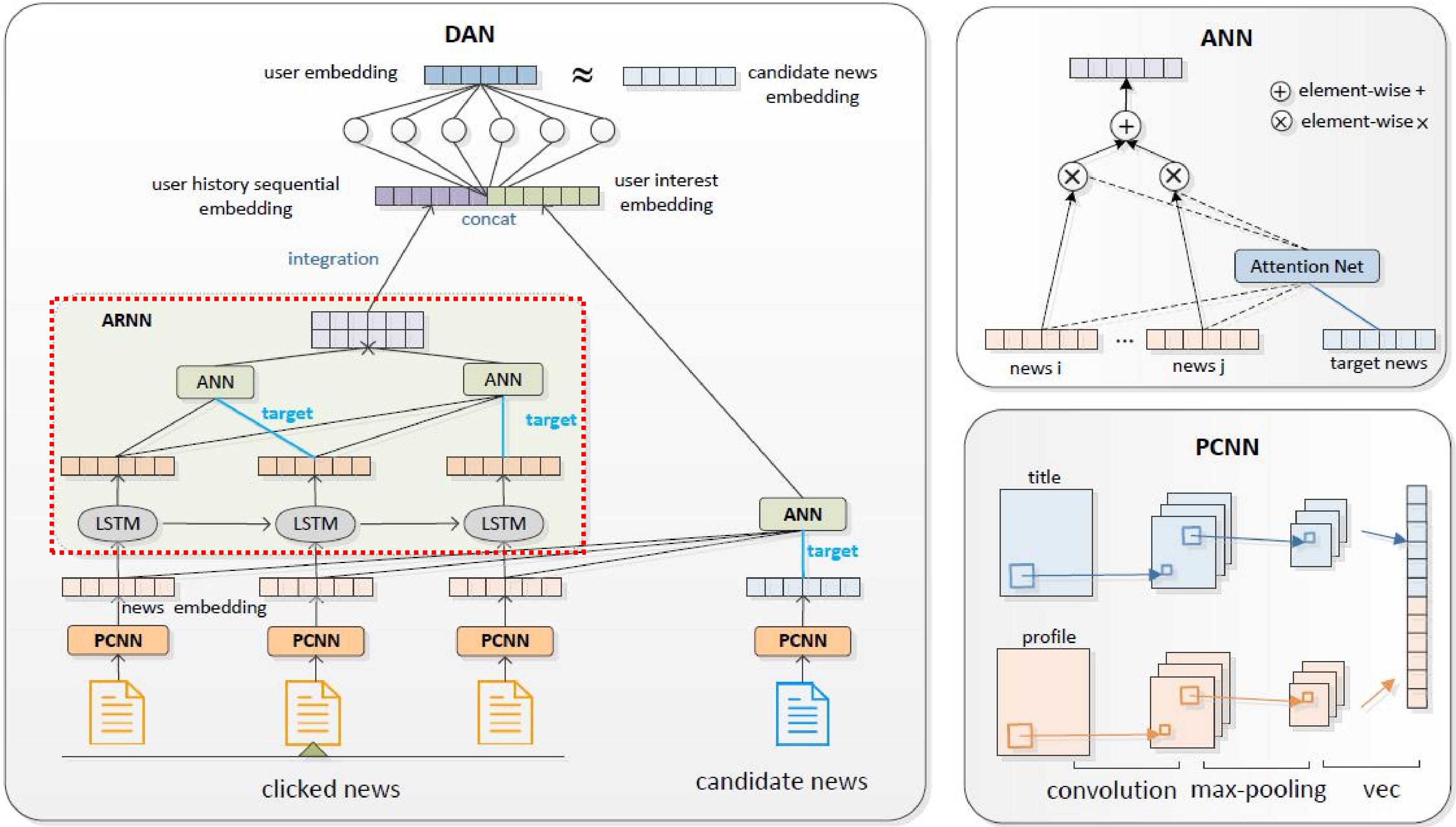
Issues of existing models

- have the cold start problem when being exposed to the sparsity of user-item interactions
- have difficulties in reflecting a user's interests in real time
- ignore the news profile
- can not consider the influence of sequential information of a user's clicked news

Proposal

DAN

➢ DAN uses three components for capturing the dynamic of news and user's interest, and recommend news to users.



PCNN Component

- **Input**: profile embedding C and title embedding T

$$\mathbf{C} = [\mathbf{e}_1, f(\mathbf{g}_1), \mathbf{e}_2, f(\mathbf{g}_2) ... \mathbf{e}_m, f(\mathbf{g}_m)]^T \qquad \mathbf{T} = [\mathbf{w}_1, \mathbf{w}_2, ... \mathbf{w}_n]^T$$

- **Two Parallel CNN**

  (1) Convolution layer $\mathbf{m} = f(\mathbf{Z} \odot \mathbf{c} + b)$

  (2) Pooling layer $\mathbf{p}=maxpooling(\mathbf{m})$

  (3) Representation layer

$$r(\mathbf{Z}) = [vec(\mathbf{p}^1); vec(\mathbf{p}^2); ... vec(\mathbf{p}^v)]$$

- **Output**: news feature representation

$$\mathbf{I} = [r(\mathbf{T}); r(\mathbf{C})]$$

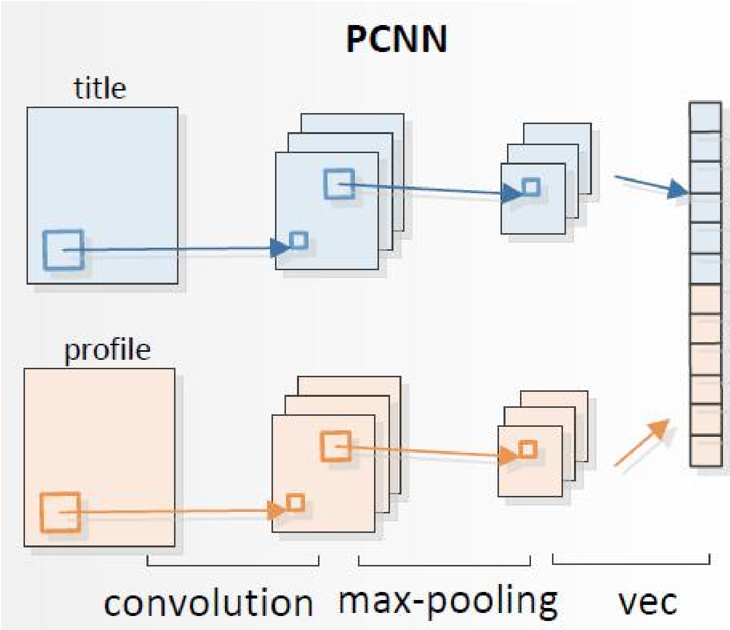


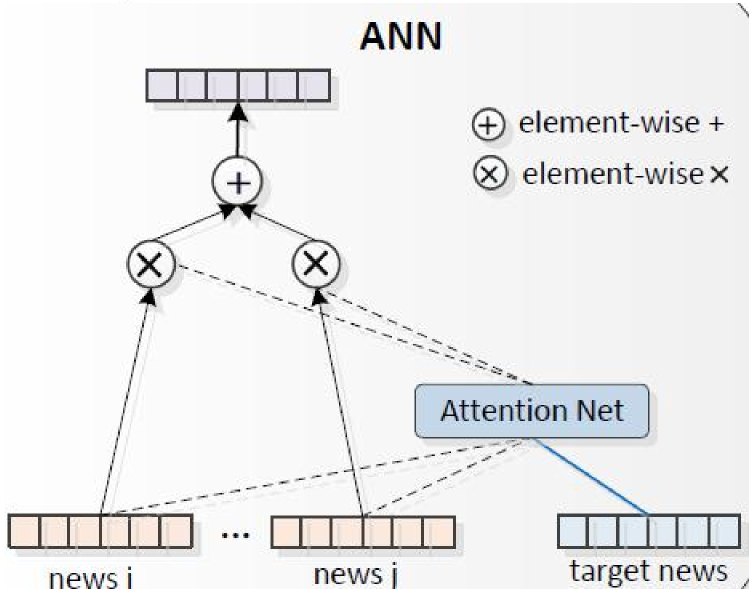
Figure 2

Proposal

ANN Component

- **Input**: clicked news representations and candidate news representation $\{\mathbf{I}_1, \mathbf{I}_2, ..., \mathbf{I}_{t-1}\}$ and $\mathbf{I}_t$
- **Attention Mechaism**

$$\mathbf{u}_j = tanh(\mathbf{W}_w\mathbf{I}_j + \mathbf{b}_w)$$
$$\mathbf{u}_t = tanh(\mathbf{W}_t\mathbf{I}_t + \mathbf{b}_t)$$
$$\alpha_{j,t} = \frac{exp(\mathbf{v}^T(\mathbf{u}_t + \mathbf{u}_j))}{\sum_j exp(\mathbf{v}^T(\mathbf{u}_t + \mathbf{u}_j))}$$
$$\mathbf{s}_t = \sum_j \alpha_{j,t}\mathbf{I}_j$$



- **Output**: user's current interest representation

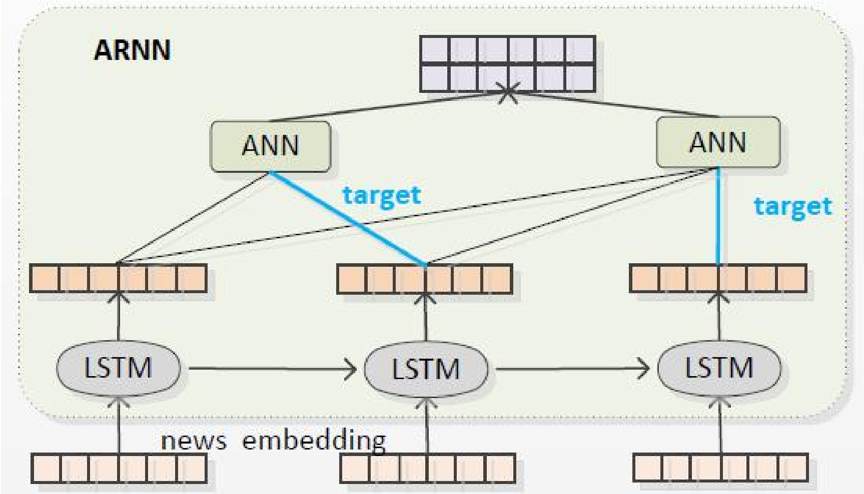
$$\mathbf{s}_t = \sum_j \alpha_{j,t}\mathbf{I}_j$$

Proposal

ARNN Component

- **Input**: clicked news representations

$$\{\mathbf{I}_1, \mathbf{I}_2, \ldots, \mathbf{I}_{t-1}\}$$



- **Attention mechanism on RNN**

(1) In j-th step, get the j-th hidden state

$$\mathbf{h}_j = LSTM(\mathbf{h}_{j-1}, \mathbf{I}_j)$$

(2) feed the first j hidden states into ANN,

and get the j-th sequencial information $\mathbf{s}_j$

(3) get the entire sequential information

$$\mathbf{S} = [\mathbf{s}_2\ \mathbf{s}_3\ \ldots \mathbf{s}_{t-1}] \qquad f(\mathbf{S}) = cnn(\mathbf{S})$$

- **Output**: user's history sequential feature representation

$$\tilde{\mathbf{h}} = f(\mathbf{S})$$

Proposal

User Feature Representation

- The concatenation of feature representations $\tilde{\mathbf{h}}$ and $\mathbf{s}_t$ is fed into a fully connected network for getting final user's embedding

$$\tilde{\mathbf{I}}_t = \mathbf{M}_g[\tilde{\mathbf{h}}; \mathbf{s}_t]$$

Similarity

- Using the similarity between user's embedding and candidate news embedding defines the probability that user clicks the candidate news x_t

$$P = cosine(\tilde{\mathbf{I}}_t \cdot \mathbf{I}_t)$$

Proposal

Training

input sample $X = (\{x_1, x_2, ..., x_{t-1}\}, x_t, y)$

- x_j is the j-th news clicked by users, x_t is the candidate news
- $y = 1$ for positive input sample $y = 0$ for the negative sample
- each input sample has the respective estimated probabilities $P \in [0, 1]$ of the user clicking the news x_t

$$L_r = -\{ \sum_{X \in \Delta +} y \, log \, P + \sum_{X \in \Delta -} (1 - y) \, log(1 - P)\} \quad (3)$$

where Δ^+ and Δ^- are the positive sample set and negative sample set.

Experiments

Data sets

Adressa is an event-based news dataset that includes anonumized users with their clicked news articles

- Adressa-1week: from 1 January to 7 January 2017
- Adreesa-10week: from 1 January to 31 March 2017

Table 1: Statistics of the dataset.

Number	*Adressa-1week*	*Adressa-10week*
#users	640,503	3,614,911
#news	20,428	81,018
#events	3,101,991	35,244,078
#entity	160,559	417,572
#entity-type	19	19
#average words per title	6.57	6.64
#average entities per news	27.7	26.5
#average entity-types per news	12.6	12.5

Adressa download from http://reclab.idi.ntnu.no/dataset/

Experiments

Result and analysis

Table 2: Comparison of different models.

Model	Adressa-1week		Adressa-10week	
	F1	AUC	F1	AUC
LibFM(-)	63.93	61.79	55.75	53.83
LibFM	70.69	69.53	64.44	61.41
DSSM(-)	69.36	68.25	62.24	60.74
DSSM	74.78	72.71	69.11	67.57
DeepWide(-)	67.39	64.83	59.98	57.98
DeepWide	73.94	71.07	67.87	66.80
DeepFM(-)	66.09	64.83	58.47	57.03
DeepFM	72.47	70.33	64.71	63.60
DMF	63.43	61.49	55.43	53.47
DKN(-)	75.38	73.45	68.57	60.57
DKN	79.97	77.24	70.39	67.53
DAN	**82.32**	**80.18**	**73.58**	**70.17**

(-) represents that removing the profile embedding from input matrix

Experiments

Discussion on different DAN variants

Table 3: Comparison among DAN variants.

Model	Adressa-1week		Adressa-10week	
	F1	AUC	F1	AUC
DAN without entity and entity-type	76.01	72.51	69.37	58.19
DAN with entity-type	77.21	74.49	70.76	65.42
DAN with entity	80.46	78.64	71.93	67.63
DAN with entity and entity-type	**82.32**	**80.18**	**73.58**	**70.17**
DAN without mapping	78.06	75.36	67.13	65.49
DAN with linear mapping	80.17	77.24	70.57	67.81
DAN with no-linear mapping	**82.32**	**80.18**	**73.58**	**70.17**
DAN without attention	78.13	75.44	70.23	68.76
DAN with attention	**82.32**	**80.18**	**73.58**	**70.17**
DAN with mul	72.46	68.54	60.39	58.73
DAN with sum	80.17	78.29	69.98	67.04
DAN with vec	79.91	77.74	69.46	66.29
DAN with cnn	**82.32**	**80.18**	**73.58**	**70.17**
DAN without ARNN	81.59	77.27	71.25	69.61
DAN with ARNN	**82.32**	**80.18**	**73.58**	**70.17**

Summary

Conclusion

- DAN considers the user's history sequential information and user's current interest together to determine whether the user clicks on the candidate news.
- DAN devises three components including news representation extractor PCNN, sequential information extractor ARNN and user interest extractor ANN.
- DAV significantly and consistently has considerable improvement over baselines, and achieves state-of-the-art performance

THANKS

Q&A