POLAR: Attention-based CNN for One-shot Personalized Article Recommendation

Zhengxiao Du, Jie Tang, Yuhui Ding

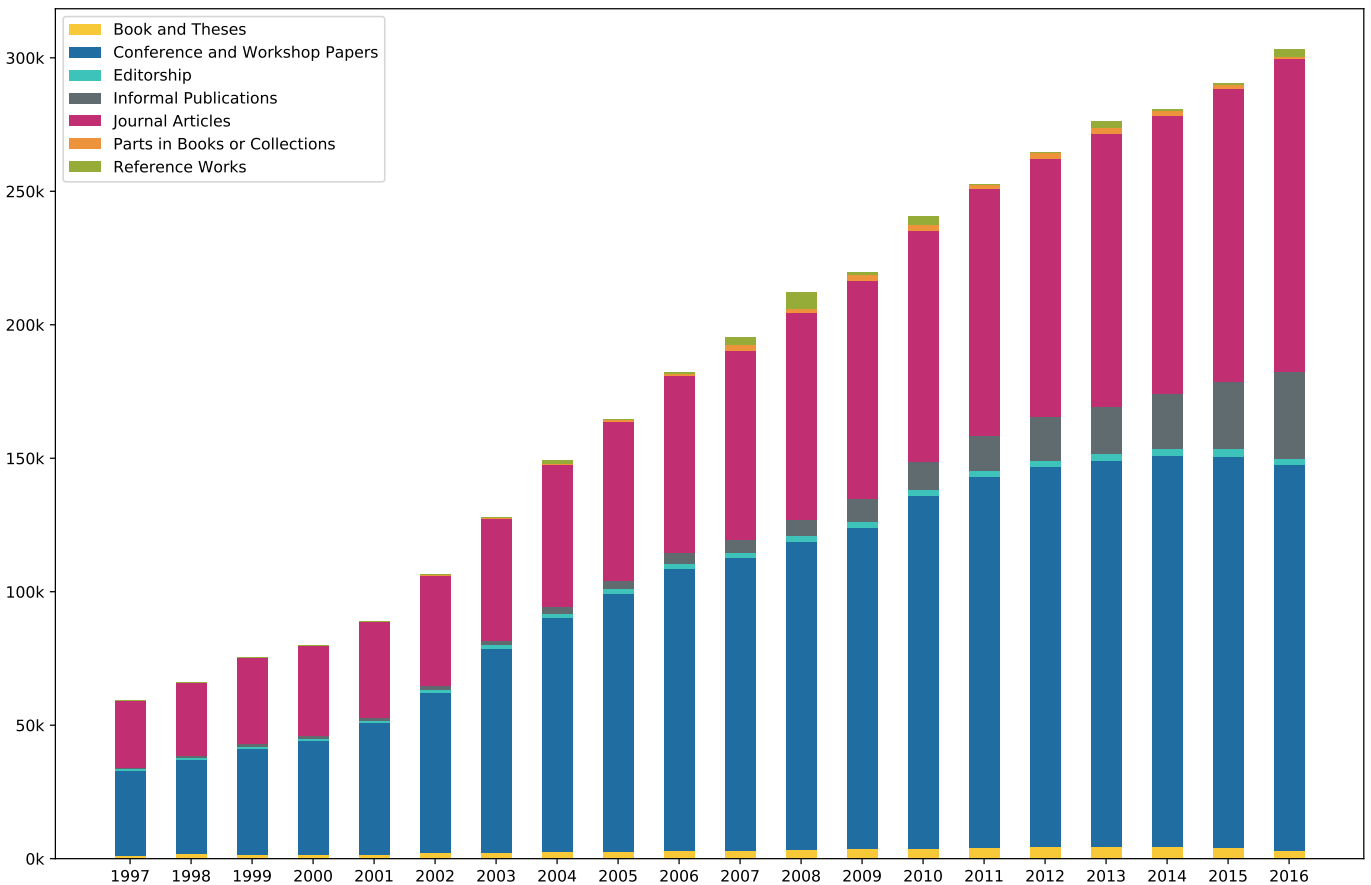
Tsinghua University

{*duzx16, dingyh15*}*@mails.tsinghua.edu.cn*,
jietang@tsinghua.edu.cn

September 13, 2018

Motivation

The publication output is growing every year
(data source: DBLP)



Related-Article Recommendation

Figure: An example from AMiner.org

Genetic Algorithms in Search, Optimization and Machine Learning EI

Abstract

From the Publisher:This book brings together - in an informal and tutorial fashion - the computer techniques, mathematical tools, and research results that will enable both students and practitioners to apply genetic algorithms to problems in many fields. Major concepts are illustrated with running examples, and major algorithms are illustrated by Pascal computer programs. No prior knowledge of GAs or genetics is assumed, and only a minimum of computer programming and mathematics background is required.

Similar Paper | Reference

1 Viorel Arnautu, and Pekka Neittaanmki. Optimal Control from Theory to Computer Programs. *Optimal Control from Theory to Computer Programs,* 2014.

2 Mitsuo Gen, and Runwei Cheng. Genetic Algorithms and Manufacturing Systems Design. *Genetic Algorithms and Manufacturing Systems Design,* 1996.

3 Masatoshi Sakawa. Genetic Algorithms and Fuzzy Multiobjective Optimization. *Genetic Algorithms and Fuzzy Multiobjective Optimization,* 2001.

4 Zbigniew Michalewicz. Genetic Algorithms Plus Data Structures Equals Evolution Programs. *Genetic Algorithms Plus Data Structures Equals Evolution Programs,* 1994.

Challenge

- How to provide personalized and non-personalized recommendation?
- How to overcome the sparsity of user feedback?
- How to utilize representative texts of articles effectively?

Problem Definition

Definition

One-shot Personalized Article Recommendation Problem

Problem Definition

Definition

One-shot Personalized Article Recommendation Problem

- **Input**: query article d_q
  candidate set $D = \{d_1, d_2, \cdots, d_N\}$
  support set $S = \{(\hat{d}_i, \hat{y}_i)\}_{i=1}^{T}$ related to user u

Problem Definition

Definition

One-shot Personalized Article Recommendation Problem

- **Input**: query article d_q
  candidate set $D = \{d_1, d_2, \cdots, d_N\}$
  support set $S = \{(\hat{d}_i, \hat{y}_i)\}_{i=1}^{T}$ related to user u
- **Output**: a totally ordered set $R(d_q, S) \subset D$ with $|R| = k$

One-shot Learning

Image Classification[1]

$$\hat{y} = \sum_{i=1}^{k} a(\hat{x}, x_i) y_i$$

[1] Vinyals et al., Matching Networks for One Shot Learning.

One-shot Learning

Image Classification[1]

$$\hat{y} = \sum_{i=1}^{k} a(\hat{x}, x_i) y_i$$

Article Recommendation

- Query article d_q
- Support set $\{(d_i, y_i)\}_{i=1}^{T}$

$$\frac{1}{T} \sum_{j=1}^{T} c(\hat{d}_i, d_j) y_j$$

- the matching to the user preference(maybe missing)

[1] Vinyals et al., Matching Networks for One Shot Learning.

One-shot Learning

Image Classification[1]

$$\hat{y} = \sum_{i=1}^{k} a(\hat{x}, x_i) y_i$$

Article Recommendation

- Query article d_q
- Support set $\{(d_i, y_i)\}_{i=1}^{T}$

$$\hat{s}_i = \frac{1}{T} \sum_{j=1}^{T} c(\hat{d}_i, d_j) y_j$$

- the matching to the user preference(maybe missing)

[1] Vinyals et al., Matching Networks for One Shot Learning.

One-shot Learning

Image Classification[1]

$$\hat{y} = \sum_{i=1}^{k} a(\hat{x}, x_i) y_i$$

Article Recommendation

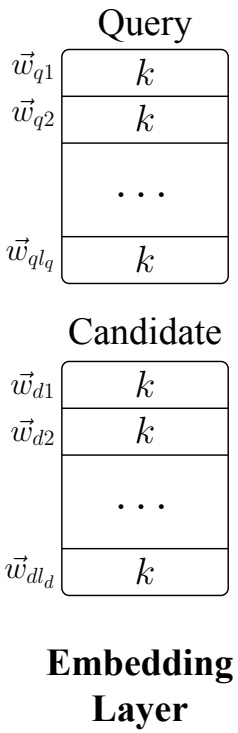
- Query article d_q
- Support set $\{(d_i, y_i)\}_{i=1}^{T}$

$$\hat{s}_i = c(d_q, \hat{d}_i) + \frac{1}{T} \sum_{j=1}^{T} c(\hat{d}_i, d_j) y_j$$

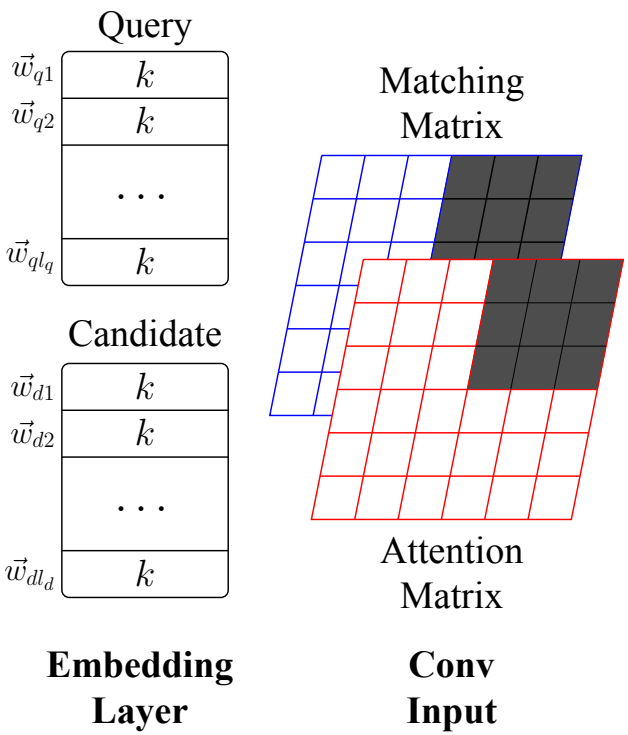
- the matching to the query article
- the matching to the user preference(maybe missing)

[1] Vinyals et al., Matching Networks for One Shot Learning.

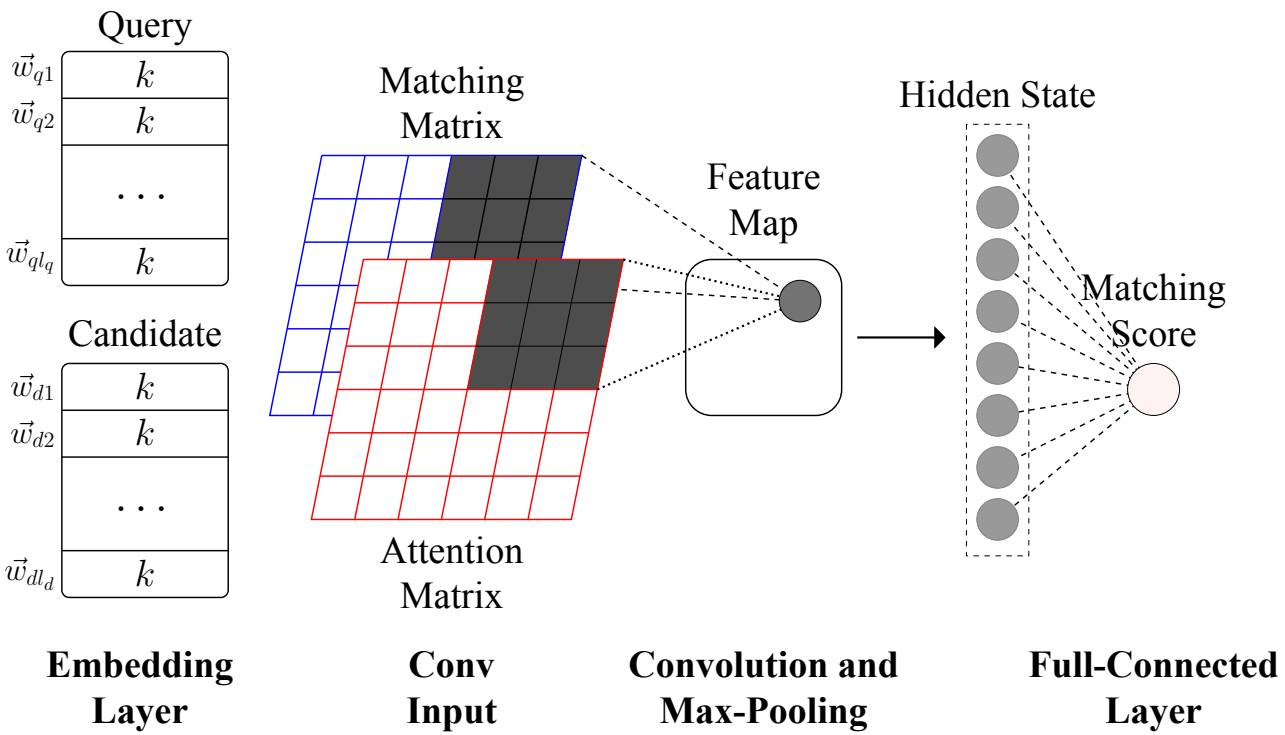
Architecture



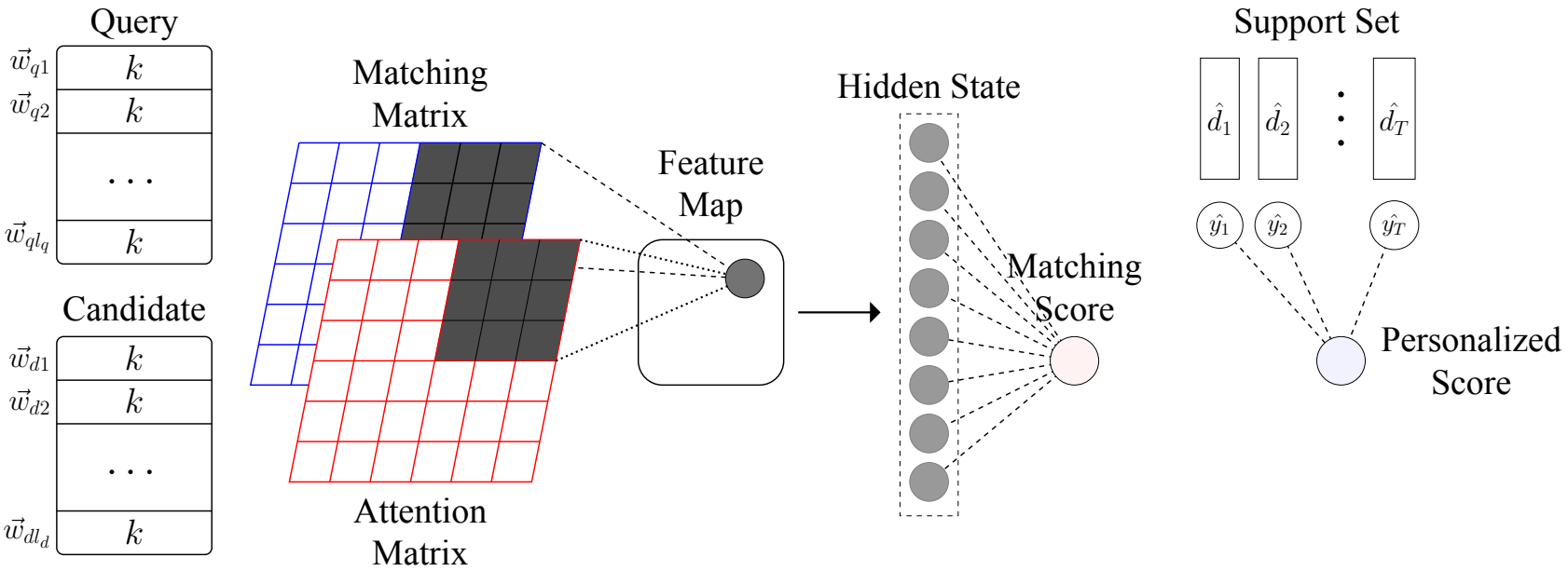
Architecture



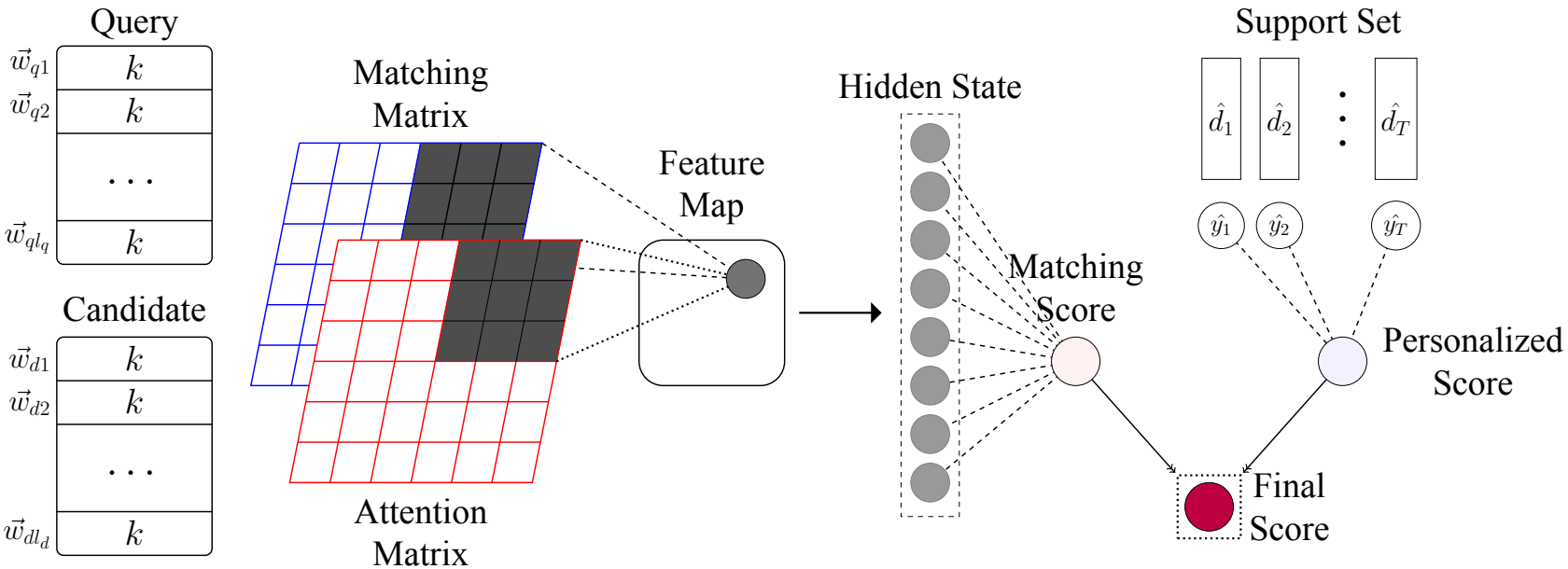
Architecture



Architecture



Architecture



Matching Matrix and Attention Matrix

- **Matching Matrix**:$(d_m, d_n) \rightarrow \mathbb{R}^{l_m \times l_n}$
  the similarity between the words of two articles.

$$\boldsymbol{M}_{i,j}^{(m,n)} = \frac{\vec{w}_{mi}^{T} \cdot \vec{w}_{nj}}{\left\|\vec{w}_{mi}\right\| \cdot \left\|\vec{w}_{nj}\right\|}$$

Matching Matrix and Attention Matrix

- **Matching Matrix**:$(d_m, d_n) \rightarrow \mathbb{R}^{l_m \times l_n}$
  the similarity between the words of two articles.

$$\boldsymbol{M}_{i,j}^{(m,n)} = \frac{\vec{w}_{mi}^{T} \cdot \vec{w}_{nj}}{\|\vec{w}_{mi}\| \cdot \|\vec{w}_{nj}\|}$$

- **Attention Matrix**:$(d_m, d_n) \rightarrow \mathbb{R}^{l_m \times l_n}$
  the importance of the matching signals

$$\boldsymbol{A}_{i,j}^{(m,n)} = r_{mi} \cdot r_{nj}$$

Local Weight and Global Weight

The word weight r_t is the product of its local weight and global weight.

Local Weight and Global Weight

The word weight r_t is the product of its local weight and global weight.

- **Global Weight**: The importance of a word in the corpus(shared among different articles)

$$v_{ij} = [\text{IDF}(t_{ij})]^{\beta}$$

Local Weight and Global Weight

The word weight r_t is the product of its local weight and global weight.

- **Global Weight**: The importance of a word in the corpus(shared among different articles)

$$v_{ij} = [\text{IDF}(t_{ij})]^{\beta}$$

The local weight is a little more complicated...

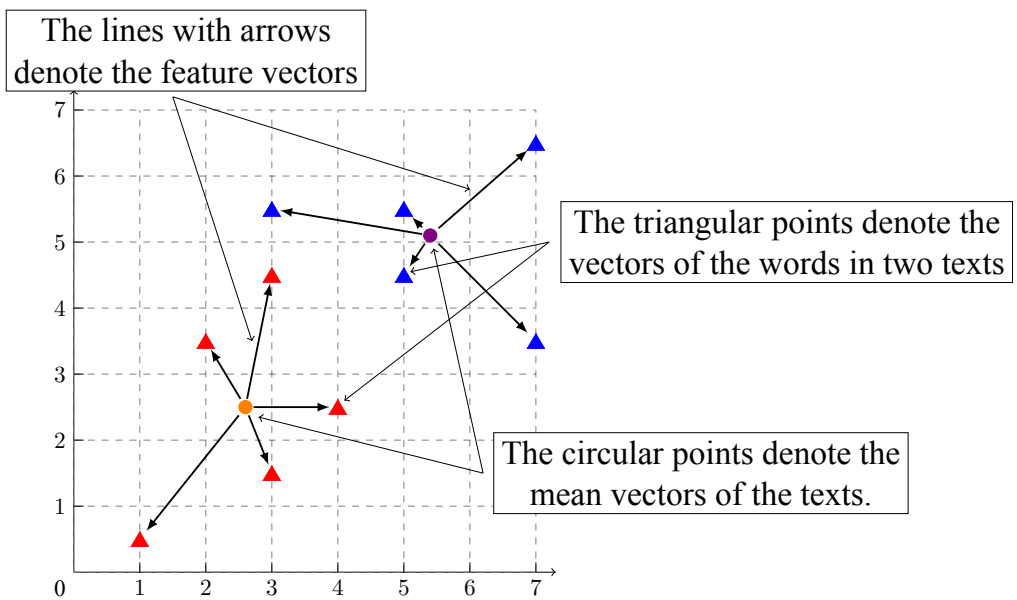
Local Weight

- **Local Weight**: The importance of a word in the article

A neural network is employed to compute the local weight.

The feature vector for word t_{ij}

$$\vec{x}_{ij} = \vec{w}_{ij} - \vec{\overline{w}}_i$$



Local Weight Network

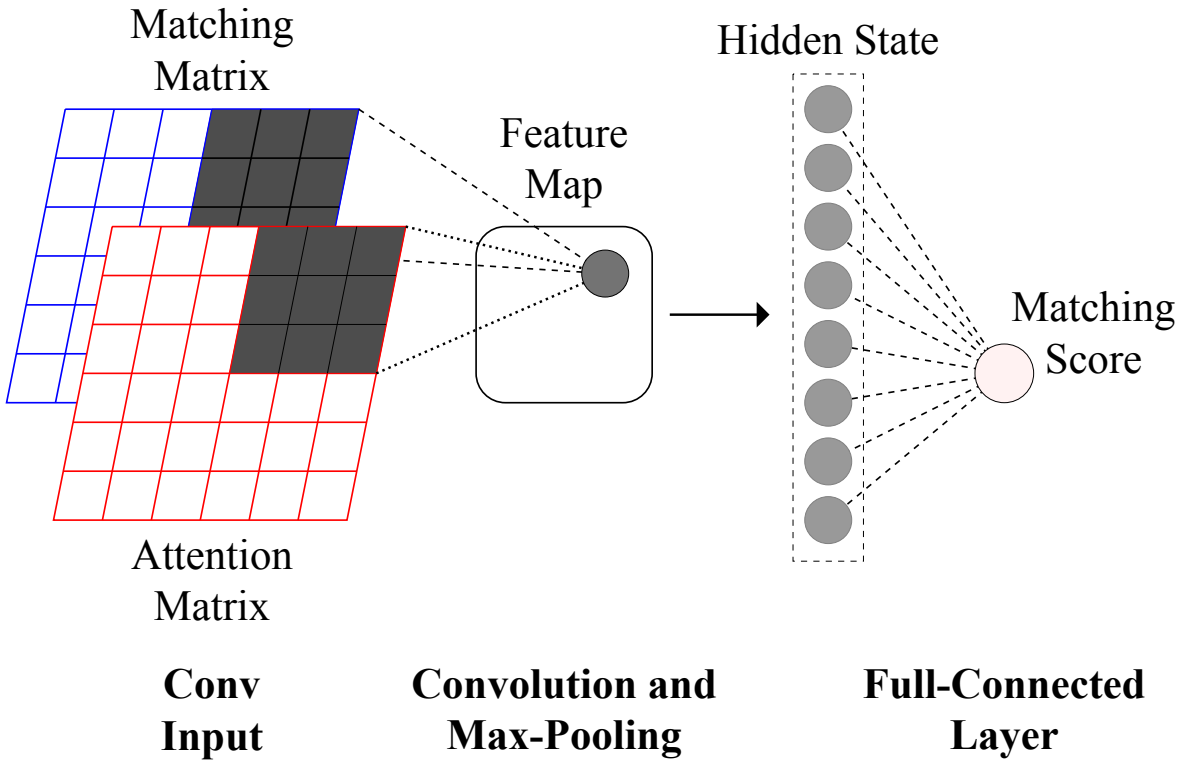
- The feature vector $\vec{x}_{ij}$ represents the semantic difference between the article and the term.
- Let $\vec{u}_{ij}^{(L)}$ be the output of the last linear layer, the output of the local weight network is

$$\mu_{ij} = \sigma(\boldsymbol{W}^{(L)} \cdot \vec{u}_{ij}^{(L)} + b^{(L)}) + \alpha$$

- α sets a lower bound for local weights.

CNN & Training

- The matching matrix and attention matrix are combined by element-wise multiplication and sent to a CNN.



- The entire model, including the local weight network, is trained on the target task.

Dataset

- **AMiner**: papers from ArnetMiner[1]
- **Patent**: patent documents from USPTO
- **RARD (Related Article Recommendation Dataset[2])**:from Sowiport, a digital library service provider.

[1] Tang et al. ArnetMiner: Extraction and Mining of Academic Social Networks. In SIGKDD'2008.

[2] Beel et al. Rard: The related-article recommendation dataset (2017)

Experimental Results

Table: Results of recommendation without personalization(%).

Method	AMiner			Patent			RARD		
	NG@3	NG@5	NG@10	NG@3	NG@5	NG@10	NG@1	NG@3	NG@5
TF-IDF	74.3	81.8	87.5	51.8	56.4	63.4	37.6	39.8	46.3
Doc2Vec	60.0	65.8	79.1	44.6	45.6	53.5	28.4	34.0	40.0
WMD	73.0	76.3	86.2	57.4	58.5	61.9	23.4	38.2	46.8
MV-LSTM	56.2	61.2	76.2	60.2	59.0	65.0	22.2	30.7	39.3
Duet	66.6	74.4	82.6	54.5	57.5	64.6	22.3	31.1	39.8
DRMM	75.0	79.9	87.1	55.0	56.2	64.7	33.1	36.3	40.6
MatchPyramid	73.5	80.0	86.8	56.4	61.4	64.4	29.1	36.2	42.8
POLAR	**80.3**	**85.2**	**90.1**	**67.8**	**69.5**	**73.6**	**42.8**	**46.3**	**51.5**

[1]For the fairness of comparison, all models don't involve personalization.
[2]NG stands for NDCG.

How One-shot Personalization Can Help

- Randomly divide the labeled articles into the support set and the candidate set to recommend.
- POLAR-OS is the proposed one-shot framework and POLAR-ALL the best model that ignores support sets in the previous part.

Table: Performance for the model with and without personalization.

Method	**AMiner**		**Patent**		**RARD**	
	NDCG@1	NDCG@3	NDCG@1	NDCG@3	NDCG@1	NDCG@3
POLAR-OS	**79.1**	**81.9**	**57.1**	**69.7**	**39.4**	**39.2**
POLAR-ALL	76.1	79.2	52.3	66.2	36.5	36.5

How Local and Global Weights Can Help

When computing the attention matrix, POLAR-LOC only uses local weights while POLAR-GLO only global weights.

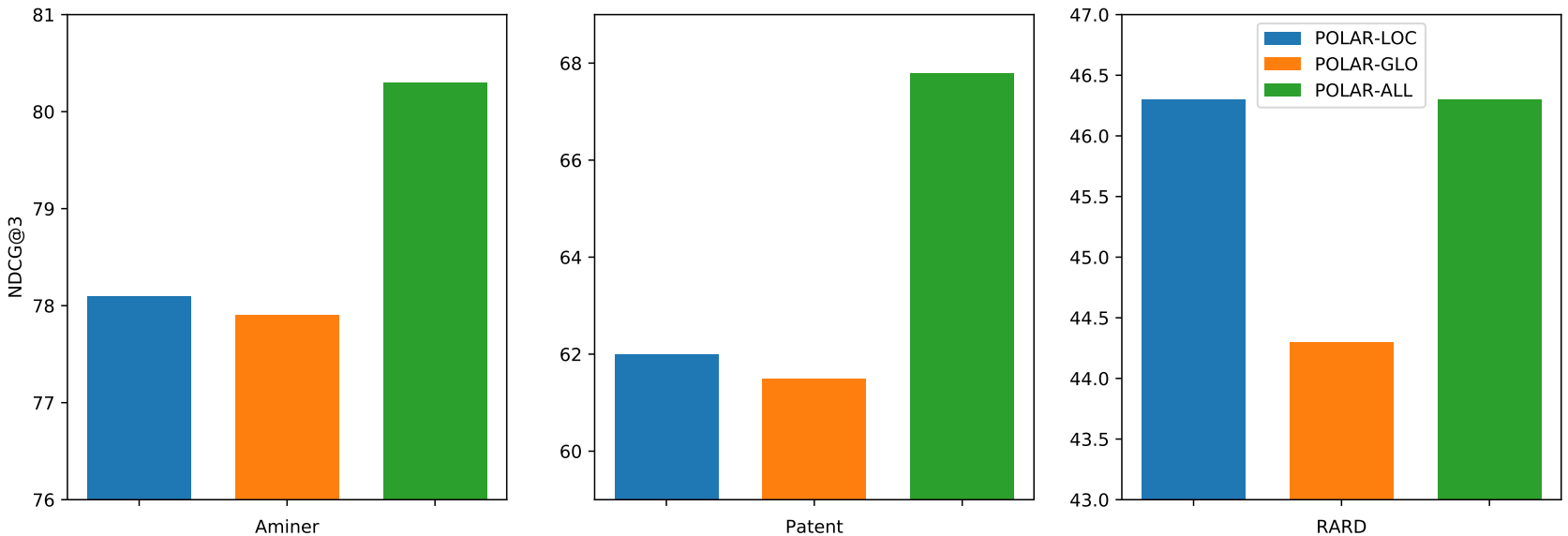


Figure: The performance of different attention matrices

Case Study: How Local and Global Weights work?

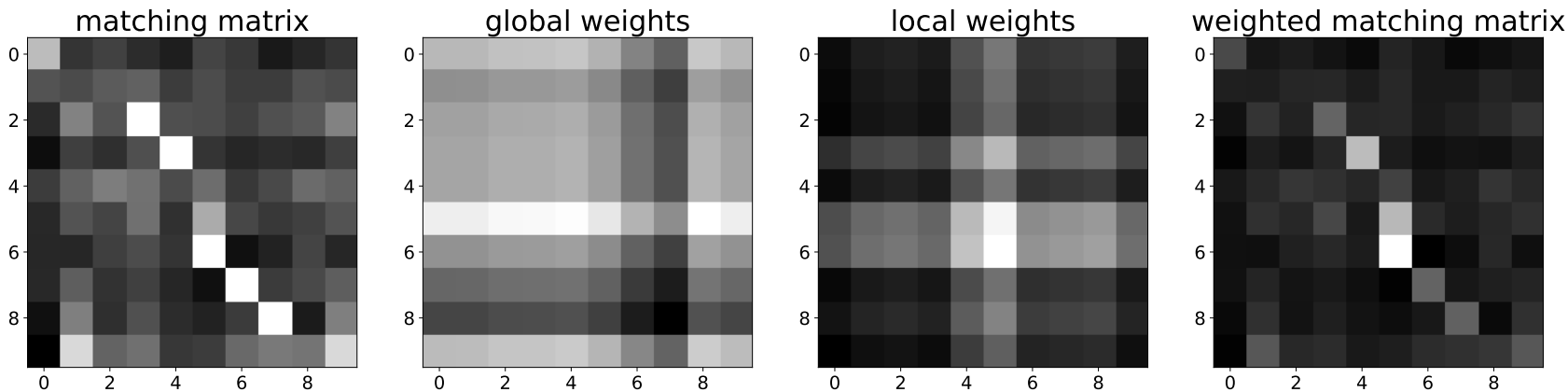


Figure: The visualization result of four matrices used in the matching of a pair of texts. The brighter the pixel is, the larger value it has.
T1:novel robust stability criteria (for) stochastic hopfield neural networks (with) time delays.
T2:new delay dependent stability criteria (for) neural networks (with) time varying delay

Case Study: How Local and Global Weights work?

Table: The statistical analysis of the local and global weights

Weight	Max	Min	Mean	Std
Local	2.00	1.00	1.20	0.15
Global	1.96	1.08	1.86	0.08

Sensitivity Analysis of Hyperparameters

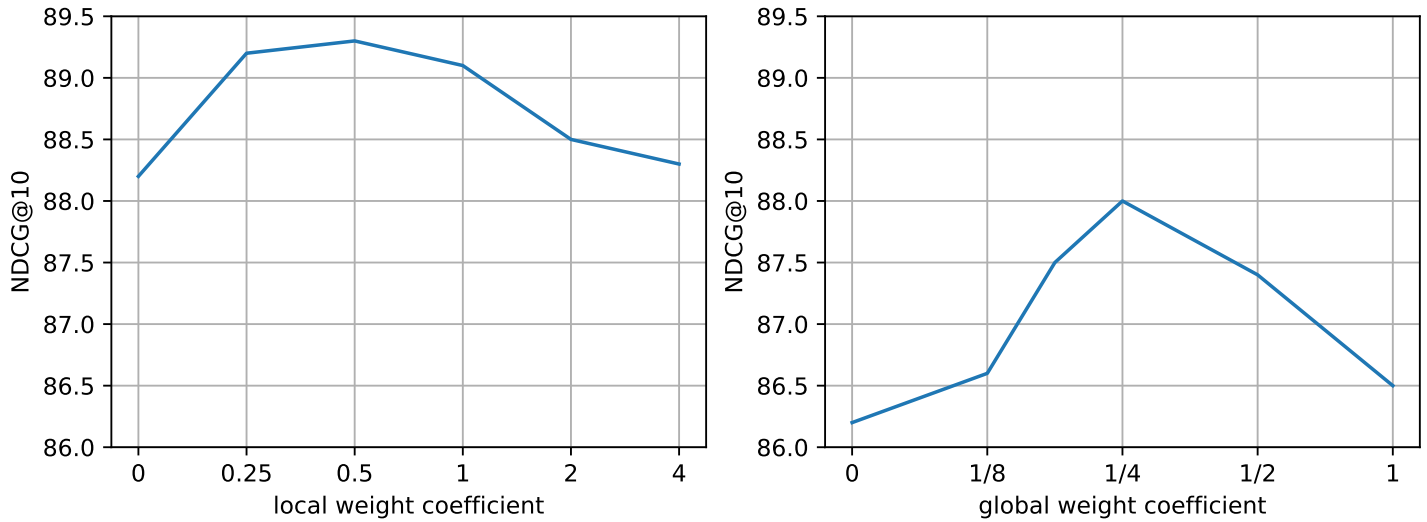


Figure: Performance comparison for POLAR-LOC with different αs and POLAR-GLO with different βs on the AMiner dataset

Conclusion

- We define the problem of one-shot personalized article recommendation.
- We utilize the framework of one-shot learning to deal with the sparse user feedback and propose an attention-based CNN for text similarity.
- We conduct experiments, whose results prove the effectiveness of the proposed model.

Any Questions?