RUCIR at NTCIR-14 WWW Task

Speaker: Jiaqing Liu

School of Information
Renmin University of China

Author: Xue Yang and **Zhicheng Dou**

Outline

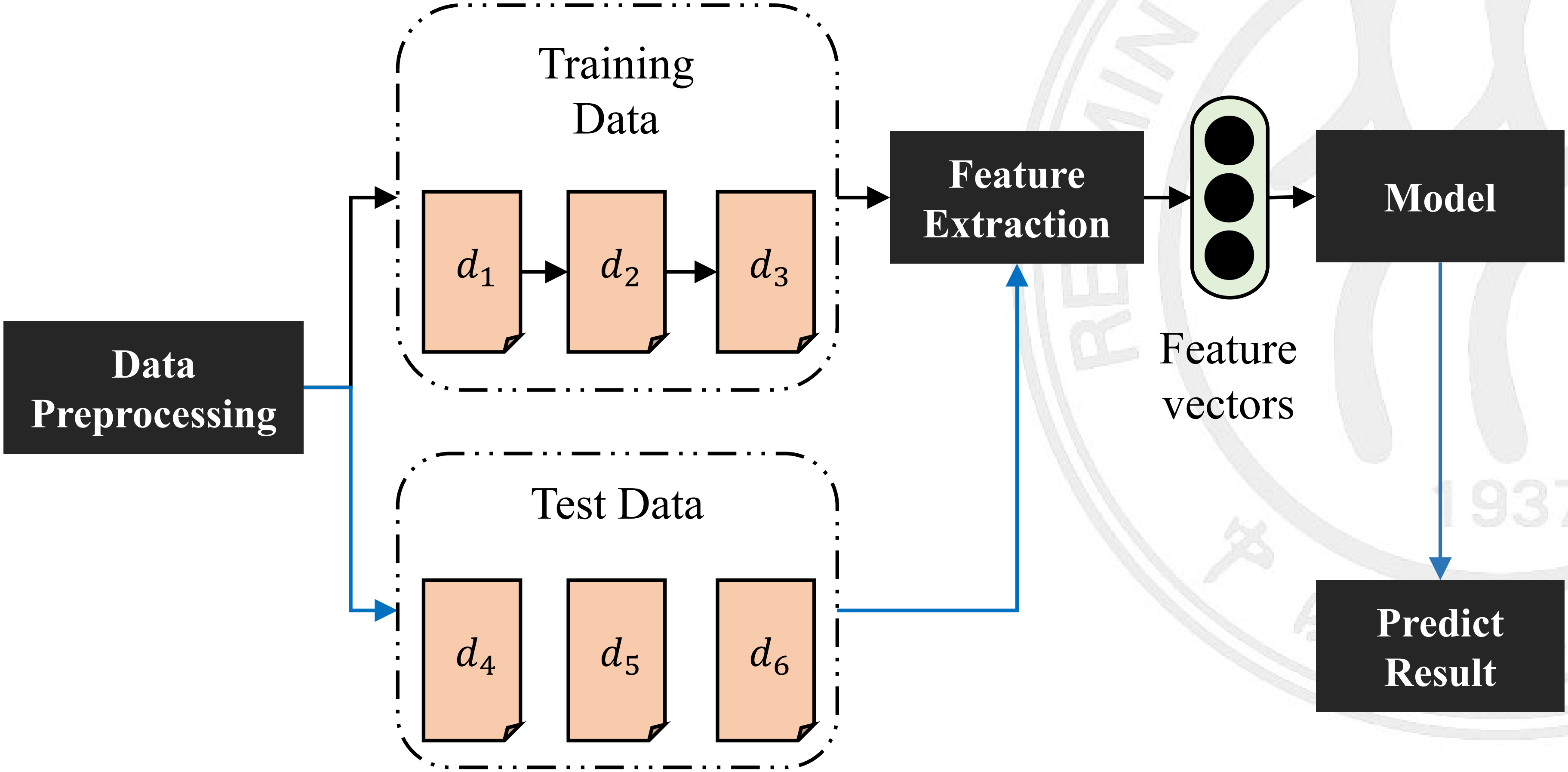
- WWW @ NTCIR-14
- Overview
- Model
- Results and analysis
- Conclusion

WWW @ NTCIR-14

- Goal: An ad hoc web search task
  - Ranking Web pages with their relevance
- Subtask 1: Chinese
  - SogouT-16 Corpus, SogouQCL Corpus
- Subtask 2: English
  - ClueWeb12-B13 Corpus

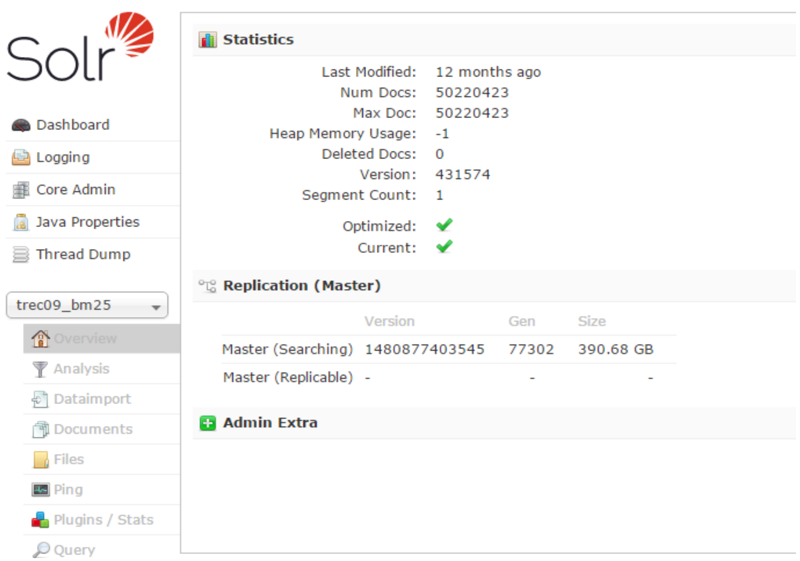
Data-Flow Overview

- Four steps



Data Preprocessing

- Pre-processing web corpus: cleaning, parsing and indexing using Solr



- Collecting official and previous TREC and NTCIR Competition labeled data for training models.
- We do not use user behavior data

Feature Extraction

- Traditional Features
  - Traditional relevance features for different fields
- Embedding Features
  - Cosine similarity between the distributed representations of query and document
- Deep Neural Features
  - Matching scores of unlabeled query-document pair by deep neural matching models

Feature Extraction (Cont'd)

- Traditional Features
  - Relevance features for four fields
    - Anchor, title, URL, and body
  - Relevance features for the whole document

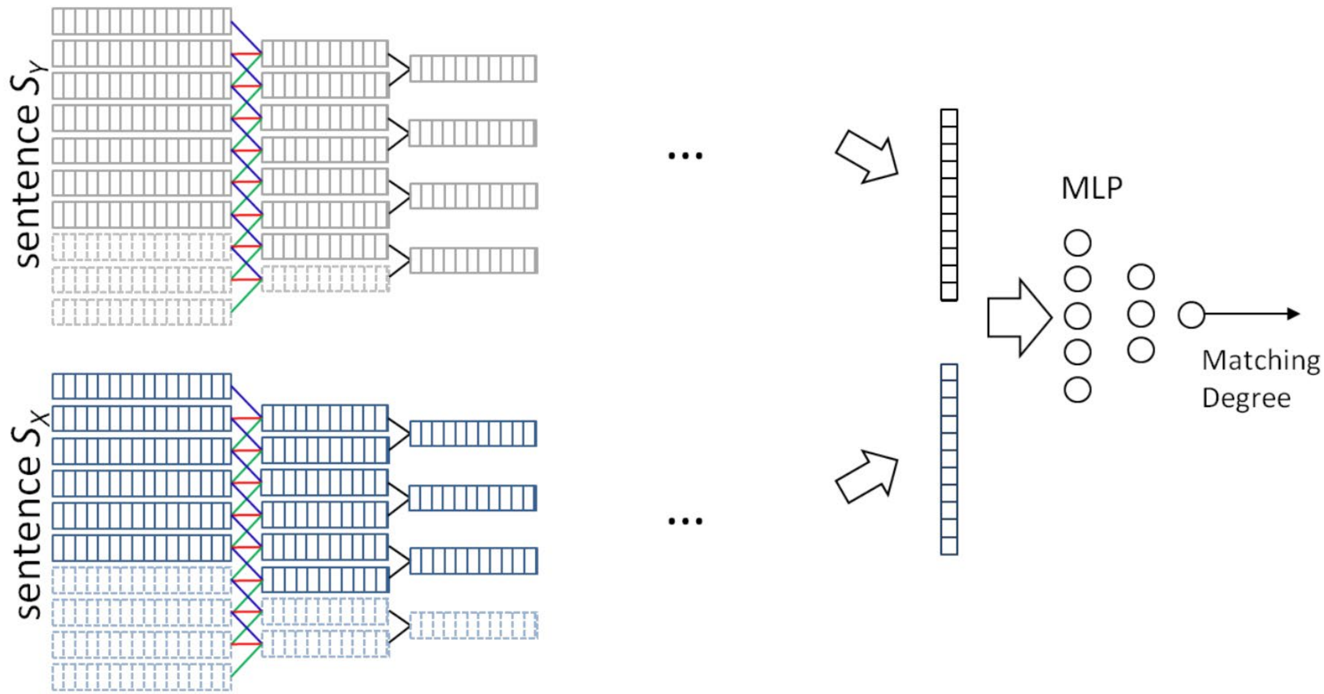
Name	Description	Fields
BM25	BM25 with default parameters	(anchor), title, URL, body, whole
TF-IDF	TF-IDF model	(anchor), title, URL, body, whole
LMIR	Language model with Dirichlet smoothing	(anchor), title, URL, body, whole
TF	Sum of term frequency	(anchor), title, URL, body, whole
IDF	Sum of inverse document frequency	(anchor), title, URL, body, whole
DL	Document length	(anchor), title, URL, body, whole
PM	Perfect match	(anchor), title, URL, body, whole
CM	Complete match	(anchor), title, URL, body, whole

Feature Extraction (Cont’d)

- Embedding Features
  - *Word2Vec* (Mikolov et al., 2013)
  - Get representations of query and document by averaging the word embedding of terms
    - $V_{di} = \frac{1}{n} \sum_{j=1}^{n} Term_{ji} , j \in [1 \dots n]$
  - Cosine similarity between query representation and document representation as feature
  - Basic use of pre-train word embedding

Feature Extraction (Cont’d)

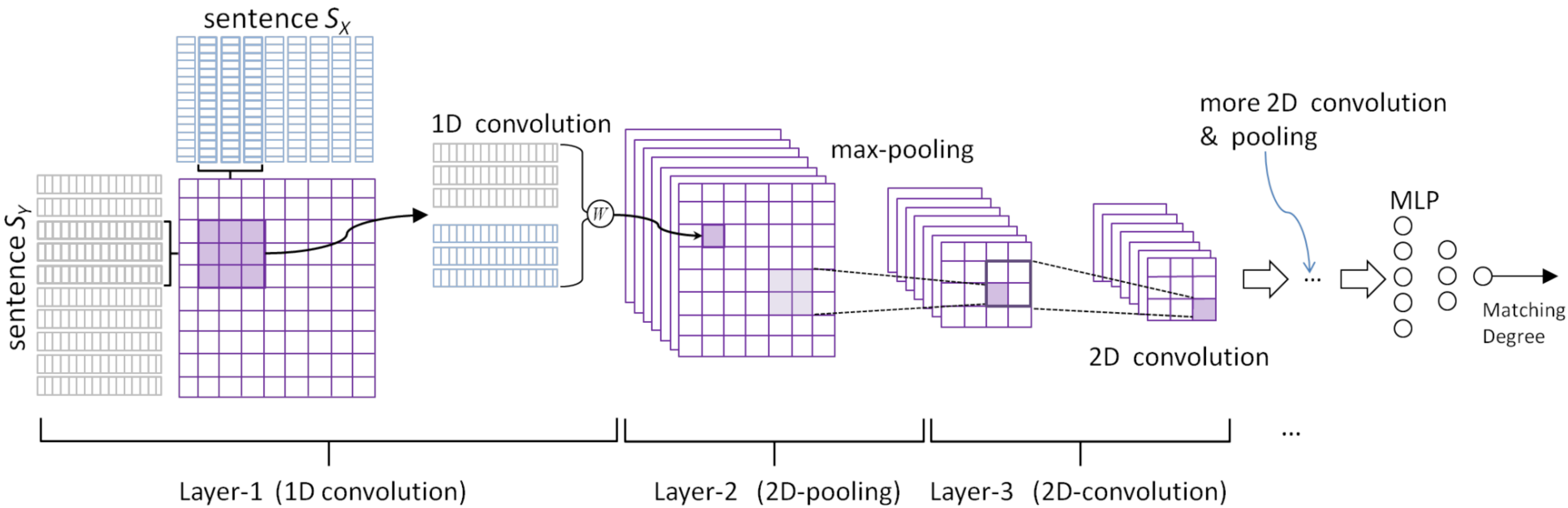
- Deep Neural Features (Matching Score)
  - *ARC-I* (Hu et al., 2014)



- Learn representation vectors of query and document with CNNs
- Get the matching score by a multi-layer perceptron layer (MLP)

Feature Extraction (Cont'd)

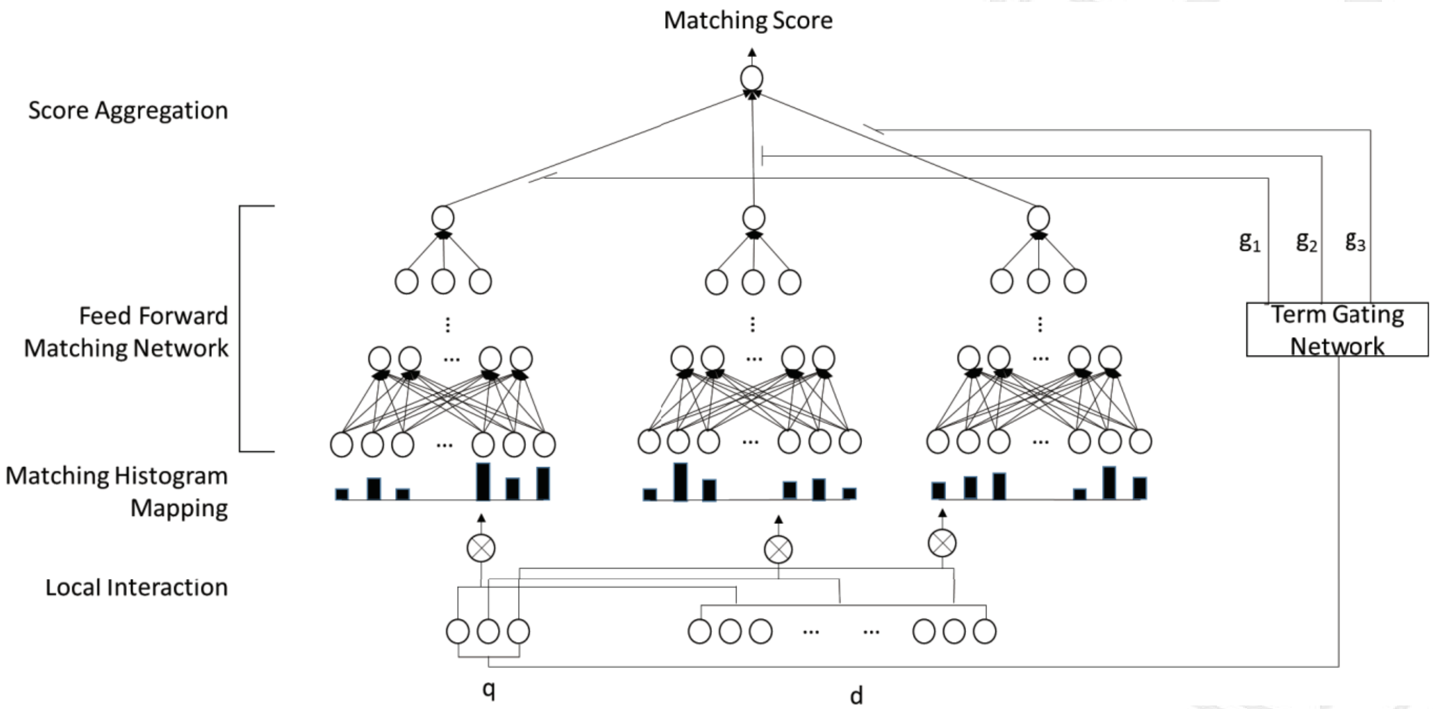
- Deep Neural Features (Matching Score)
  - *ARC-II* (Hu et al., 2014)



- Learn interaction representation vectors for query and document
- Get the matching score by a MLP after 2D pooling and convolution

Feature Extraction (Cont'd)

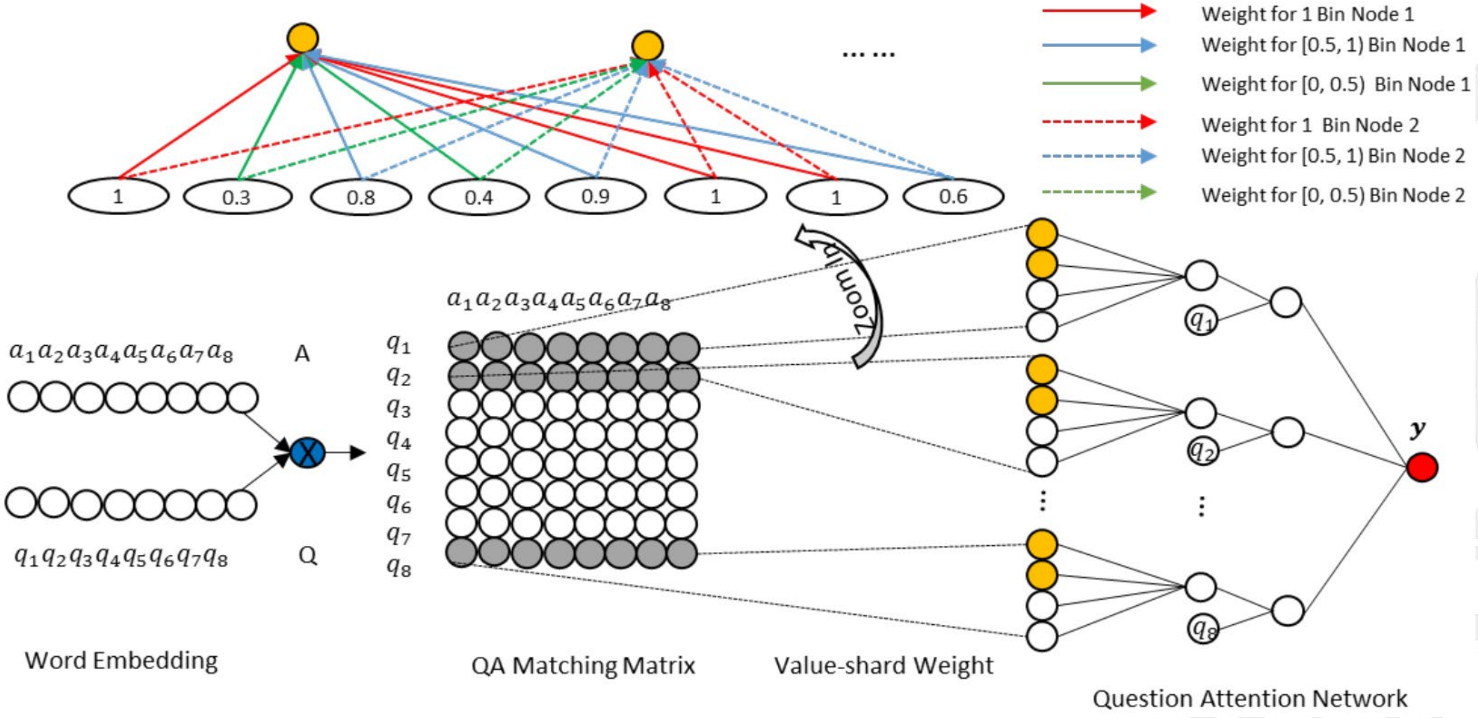
- Deep Neural Features (Matching Score)
  - *DRMM* (Guo et al., 2016)



- Matching histograms: interaction between query term with document
- Matching score: based on MLP and calculated by a softmax function

Feature Extraction (Cont'd)

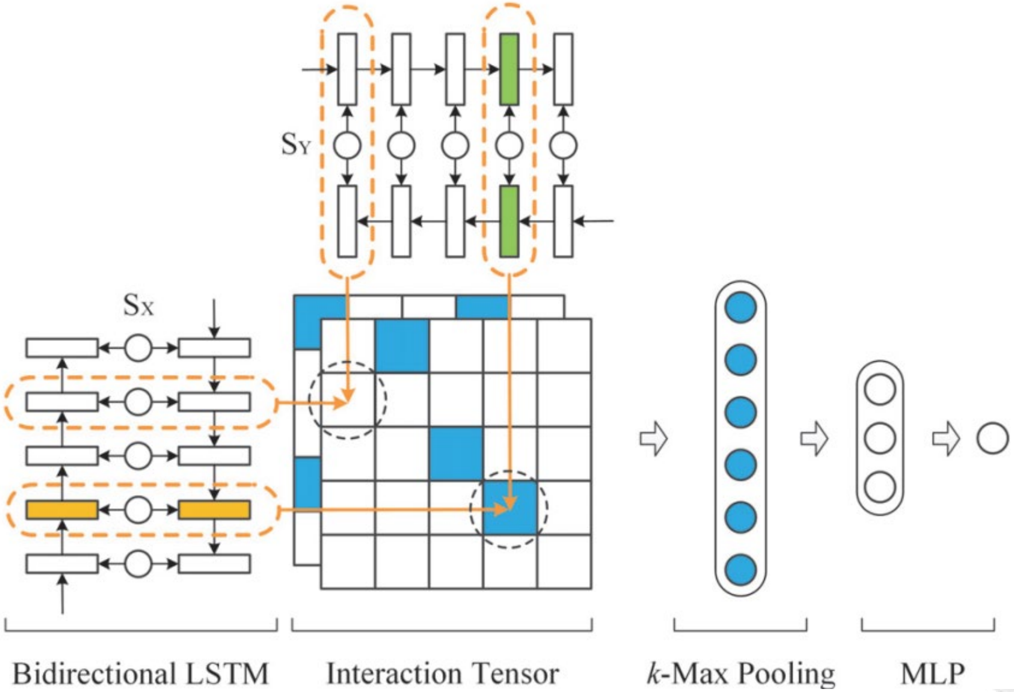
- Deep Neural Features (Matching Score)
  - *aNMM* (Yang et al., 2016)



  - Use value-shared weighting rather than position-shared (*ARC-II*)
  - Integrate the results of each query term with a softmax function

Feature Extraction (Cont'd)

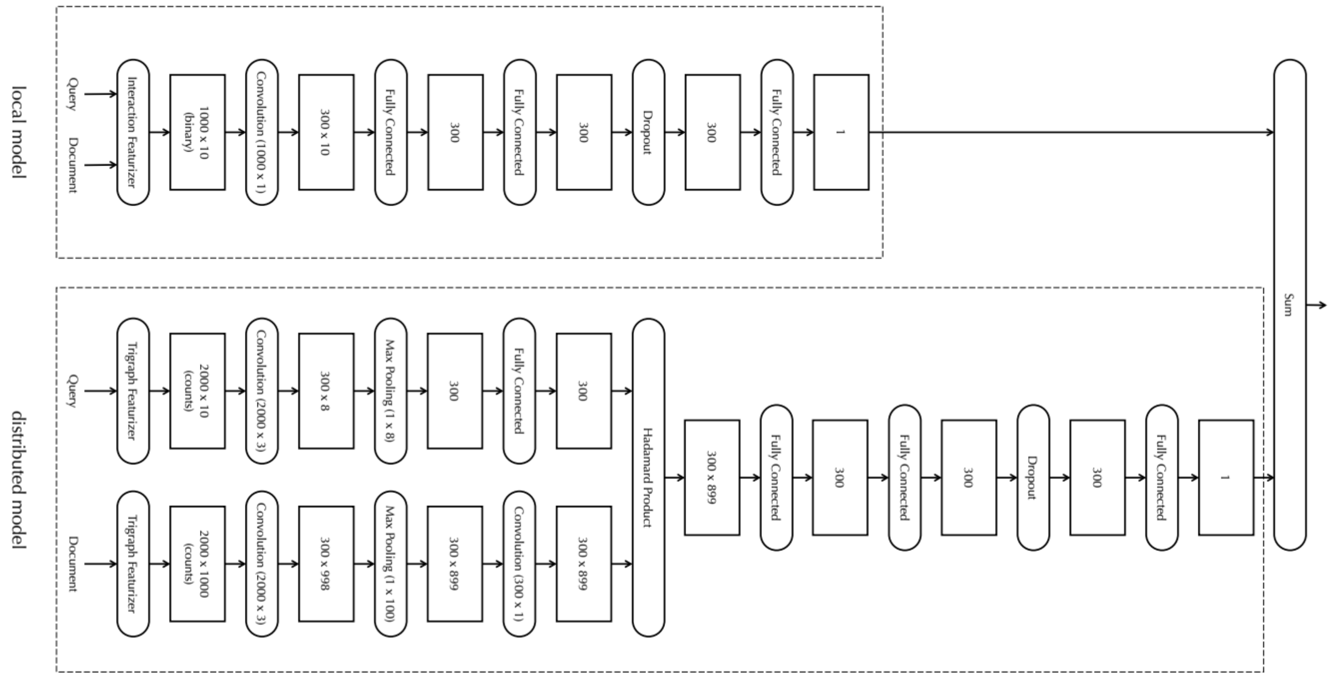
- Deep Neural Features (Matching Score)
  - *MV-LSTM* (Wan et al., 2016)



- Learn representation of query and document by bi-LSTMs
- Build interaction matrix with cosine and get score by MLP

Feature Extraction (Cont'd)

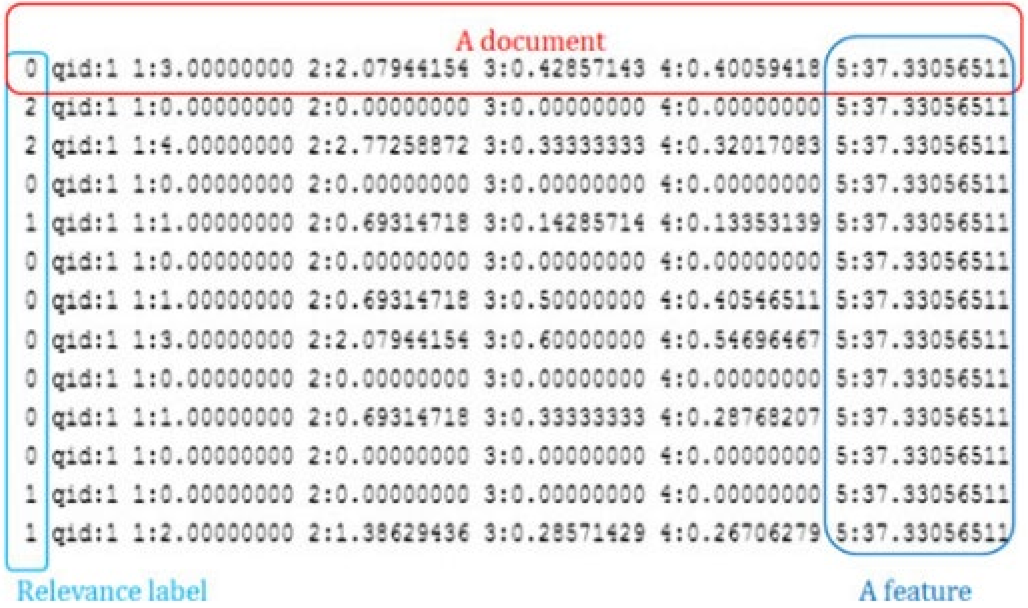
- Deep Neural Features (Matching Score)
  - *DUET* (Mitra et al., 2017)



  - Local representations: one-hot encoding to exact term match
  - Distributed representations: latent embedding based topic model

Model Training

- Input Format


A document

```
0 qid:1 1:3.00000000 2:2.07944154 3:0.42857143 4:0.40059418 5:37.33056511
2 qid:1 1:0.00000000 2:0.00000000 3:0.00000000 4:0.00000000 5:37.33056511
2 qid:1 1:4.00000000 2:2.77258872 3:0.33333333 4:0.32017083 5:37.33056511
0 qid:1 1:0.00000000 2:0.00000000 3:0.00000000 4:0.00000000 5:37.33056511
1 qid:1 1:1.00000000 2:0.69314718 3:0.14285714 4:0.13353139 5:37.33056511
0 qid:1 1:0.00000000 2:0.00000000 3:0.00000000 4:0.00000000 5:37.33056511
0 qid:1 1:1.00000000 2:0.69314718 3:0.50000000 4:0.40546511 5:37.33056511
0 qid:1 1:3.00000000 2:2.07944154 3:0.60000000 4:0.54696467 5:37.33056511
0 qid:1 1:0.00000000 2:0.00000000 3:0.00000000 4:0.00000000 5:37.33056511
0 qid:1 1:1.00000000 2:0.69314718 3:0.33333333 4:0.28768207 5:37.33056511
0 qid:1 1:0.00000000 2:0.00000000 3:0.00000000 4:0.00000000 5:37.33056511
1 qid:1 1:0.00000000 2:0.00000000 3:0.00000000 4:0.00000000 5:37.33056511
1 qid:1 1:2.00000000 2:1.38629436 3:0.28571429 4:0.26706279 5:37.33056511
```

Relevance label

A feature

- Model
  - Ranklib: LambdaMART

Evaluation Metrics

- *nDCG@K*
  - $nDCG@K = N_K^{-1} \sum_{i=1}^{n} g(r_i) d(i)$
- *Q@K*
  - $Q@K = \frac{1}{\min(K,R)} \sum_{r=1}^{k} J(r) \frac{C(r)+\beta cg(r)}{r+\beta cg^*(r)}$
- *nERR@K*
  - $nERR@K = \sum_{r=1}^{K} \frac{1}{r} \prod_{i=1}^{r-1} (1 - R_i) R_r$

Results (Chinese)

Run	Query	Features	nDCG@10	Q@10	nERR@10
RUCIR-1	Description	Traditional Embedding	0.4515	0.4228	0.5792
RUCIR-2	Content	Traditional Embedding	**0.4866**	**0.4571**	**0.6044**
RUCIR-3	Description	Traditional	0.4503	0.4223	0.5630
RUCIR-4	Description	Traditional Deep Neural	0.4458	0.4226	0.5619
RUCIR-5	Description	Deep Neural	0.2745	0.2404	0.3832

Results (English)

Run	Query	Features	nDCG@10	Q@10	nERR@10
RUCIR-1	Description	Traditional	0.3137	0.2973	0.4469
RUCIR-2	Content	Traditional	**0.3489**	**0.3352**	**0.4917**
RUCIR-3	Description	Traditional Embedding	0.3137	0.2973	0.4469
RUCIR-4	Description	Traditional Deep Neural	0.3293	0.3094	0.4602
RUCIR-5	Description	Deep Neural	0.2876	0.2659	0.4188

Analysis

- [CN & EN] Query content Run > Query description Run (CO > DE)
- [CN] Traditional features Run + Embedding features Run > Other Runs (1 > 3, 4, 5)
- [EN] Traditional features Run + Deep neural features Run > Other Runs (4 > 1, 3, 5)
- [CN & EN] Deep neural features Run << Other Runs (5 << 1, 2, 3, 4)

Conclusion

- We Want Web task
  - Matching with query content is better than matching with query description
  - Traditional text relevance features are still stable and effective
  - Using embedding feature can help a little
  - Using deep neural features can help but less than expectation, which needs future research

Thanks

Speaker: Jiaqing Liu

Author: Xue Yang and Zhicheng Dou

Email: {ruc_yangx, dou}@ruc.edu.cn