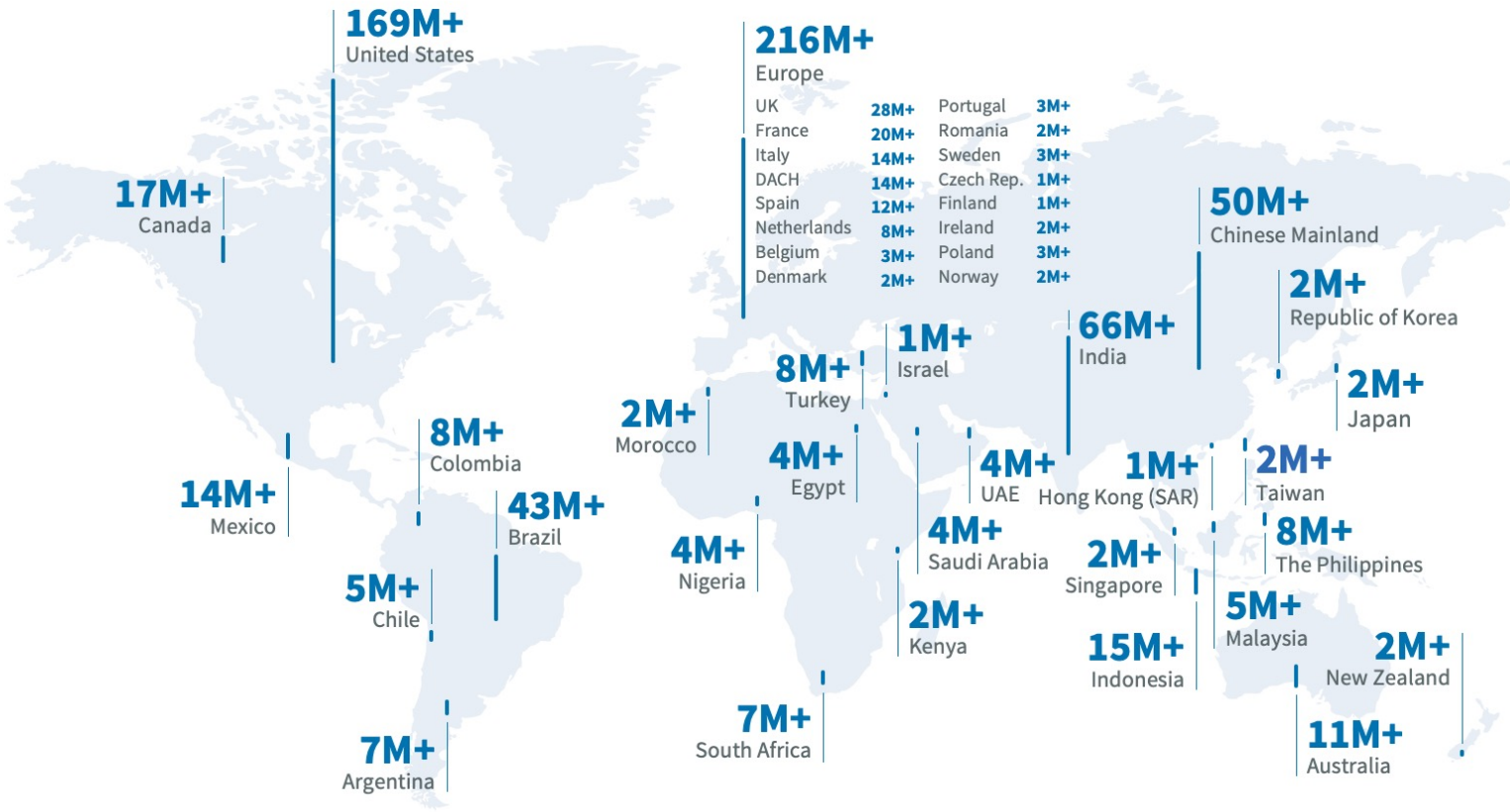
Leveraging Search History for Improving Person-Job Fit

Yupeng Hou[1], Xingyu Pan[2], Wayne Xin Zhao[1,✉], Shuqing Bian[2],
Yang Song[3], Tao Zhang[3], Ji-Rong Wen[1,2]

1. Gaoling School of Artificial Intelligence, Renmin University of China
2. School of Information, Renmin University of China
3. BOSS Zhipin

Background: online recruitment

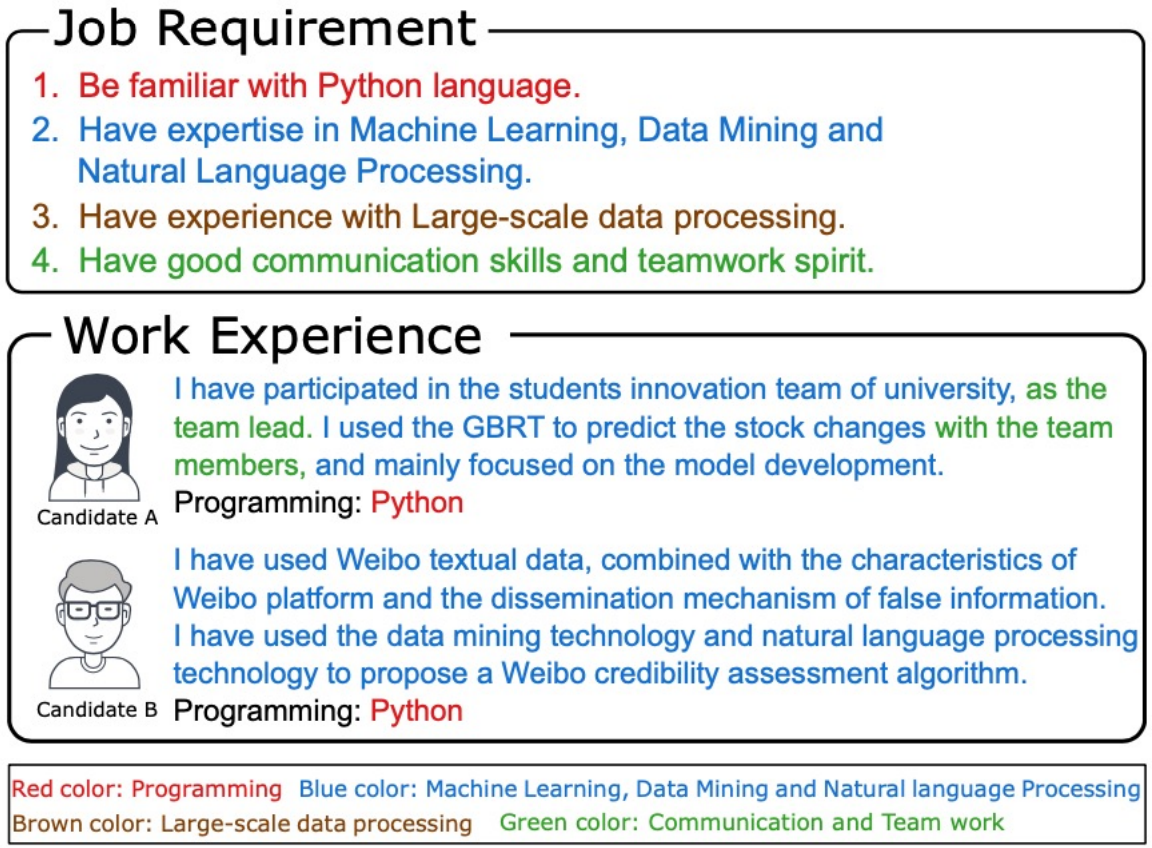
600M Candidates
20M Jobs



https://news.linkedin.com/about-us#statistics

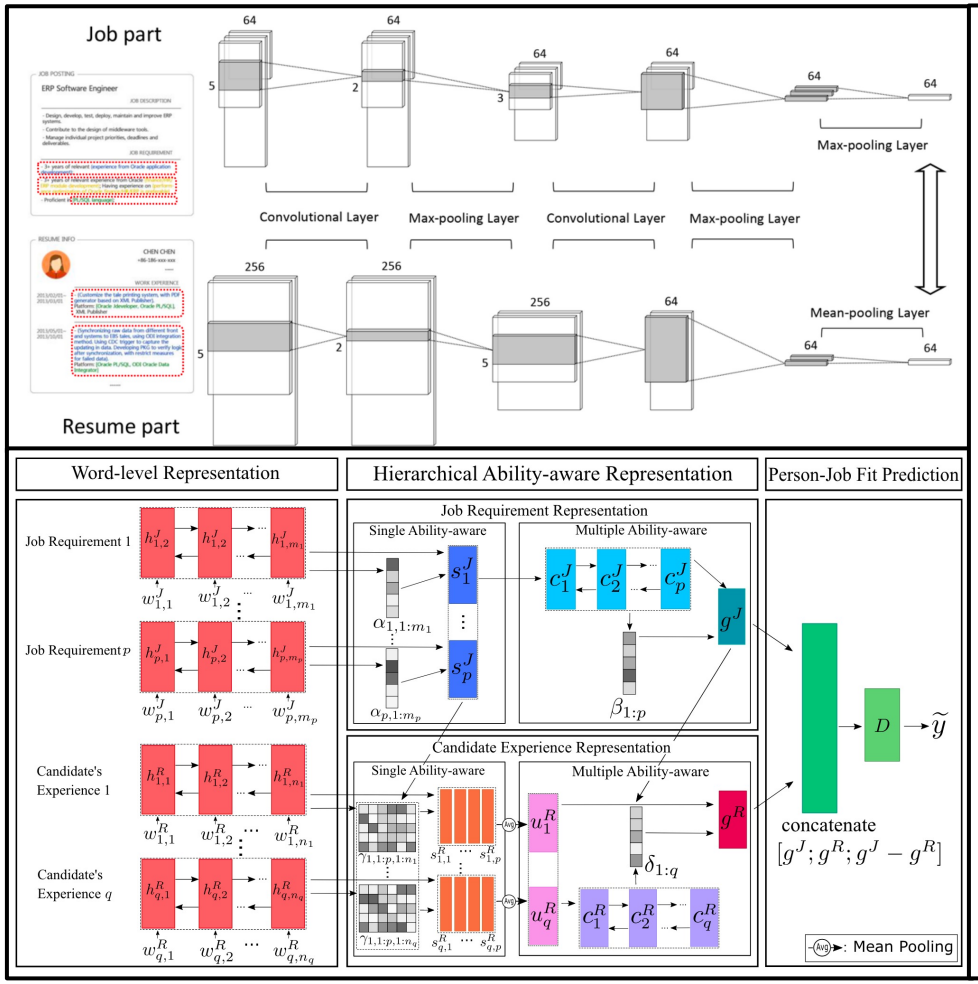
Task: *Person-Job Fit*

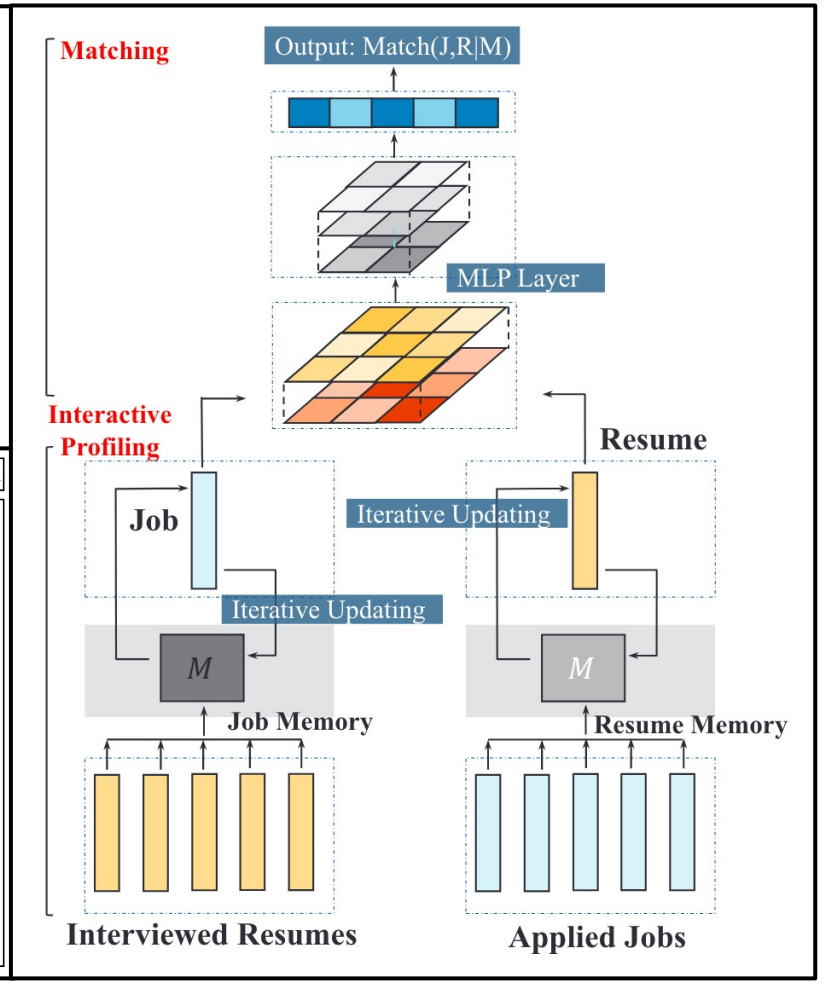
Match qualified candidates
with suitable job positions



Qin et al. "Enhancing person-job fit for talent recruitment: An ability-aware neural network approach." SIGIR 2018.

Related works





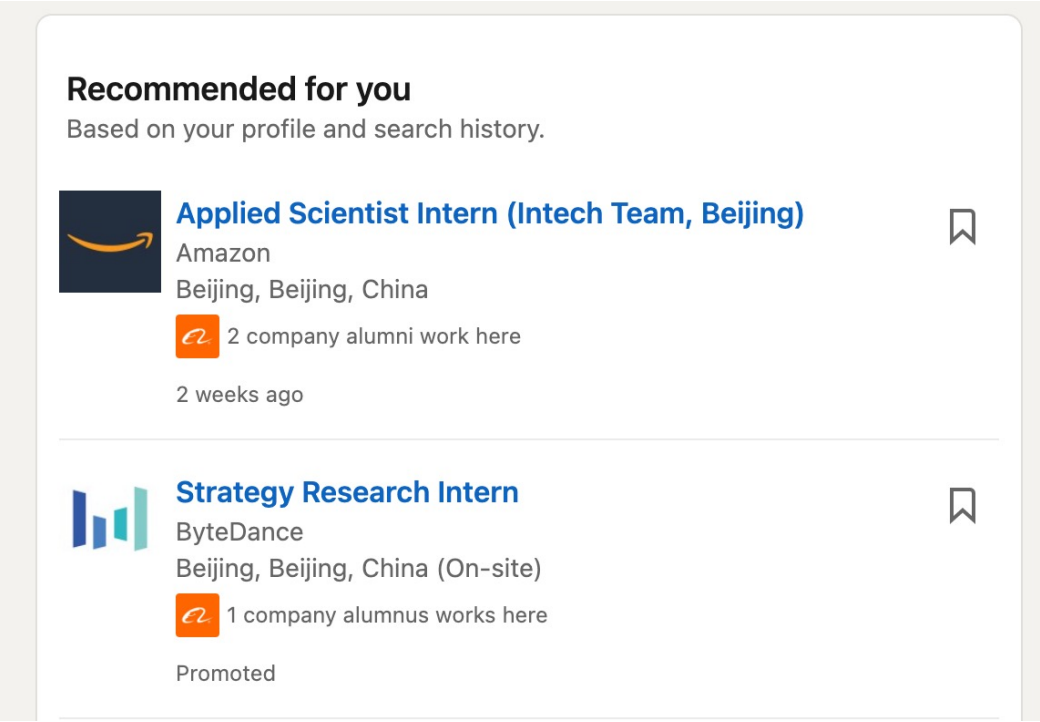
Text matching
(NN as text encoders)

Matching records from recommendation scenario as supervision signals

Zhu et al. "Person-Job Fit: Adapting the Right Talent for the Right Job with Joint Representation Learning." TMIS 2018.
Qin et al. "Enhancing person-job fit for talent recruitment: An ability-aware neural network approach." SIGIR 2018.
Yan et al. "Interview Choice Reveals Your Preference on the Market: To Improve Job-Resume Matching through Profiling Memories." KDD 2019.

Problem

- Two main scenarios, both exists interactions, text, ……



Recommendation (~81%)

Search (~19%)

Statistics come from our experimental dataset

Problem

- Two main scenarios, both exists interactions, text, ……

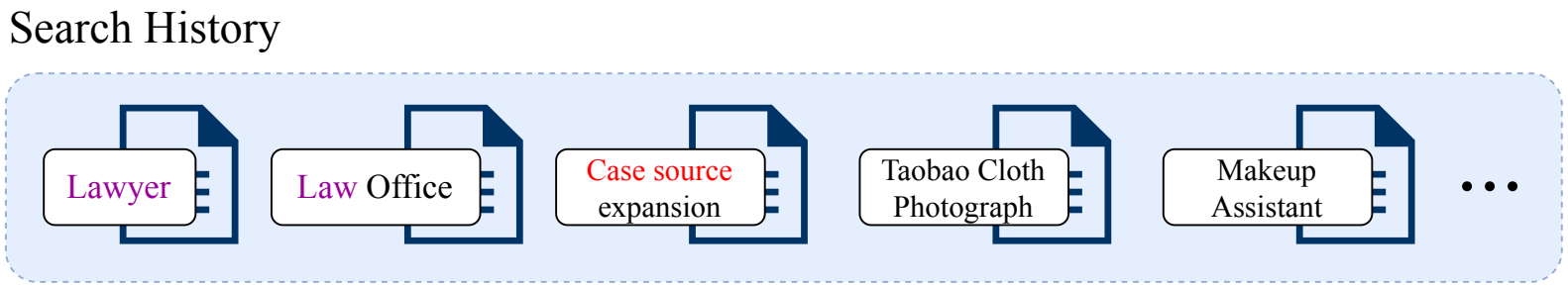
- *Search*
  - has seldom been considered 😔
  - reflects important evidence for job intention of users 🤔

Problem

- Two main scenarios, both exists interactions, text, ……
- *Search*
  - has seldom been considered 😔
  - reflects important evidence for job intention of users 🤔
- Can we leverage these two sources simultaneously? 💡

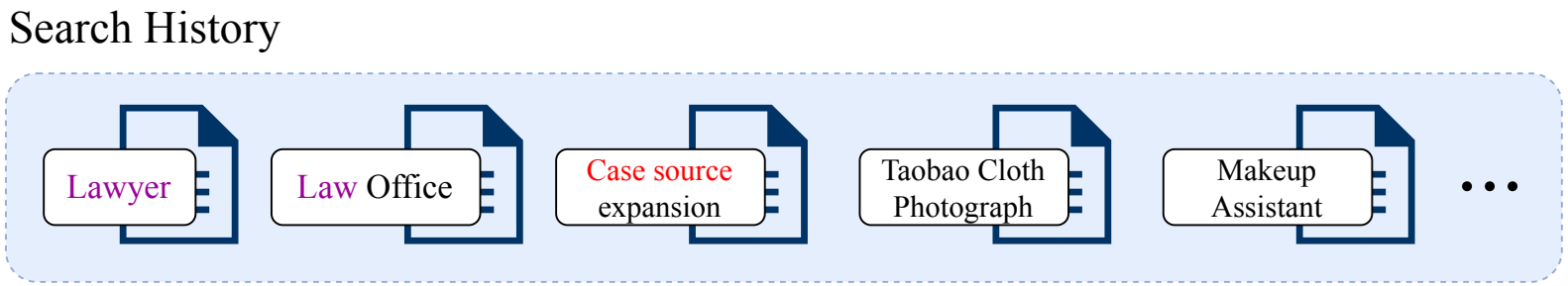
Challenges

- Hard to capture job needs from search history



Challenges

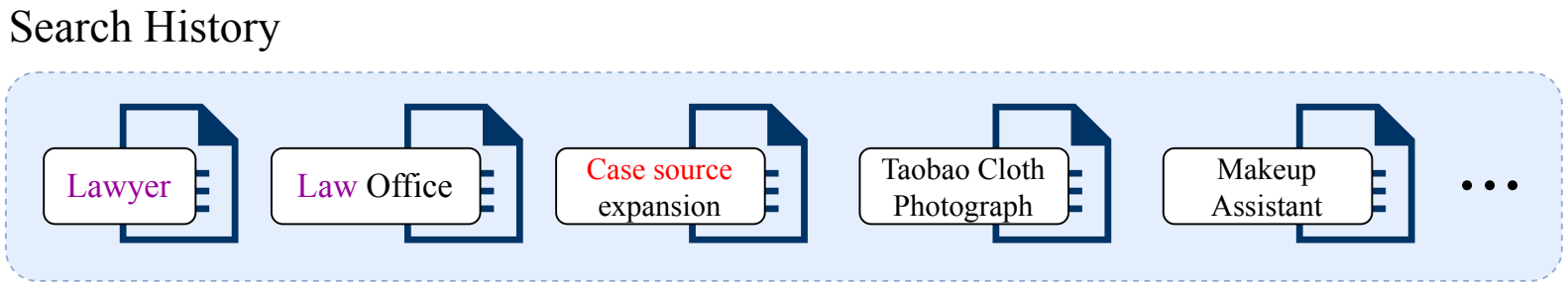
- Hard to capture job needs from search history



  - Data format: list of short queries together with clicked jobs;

Challenges

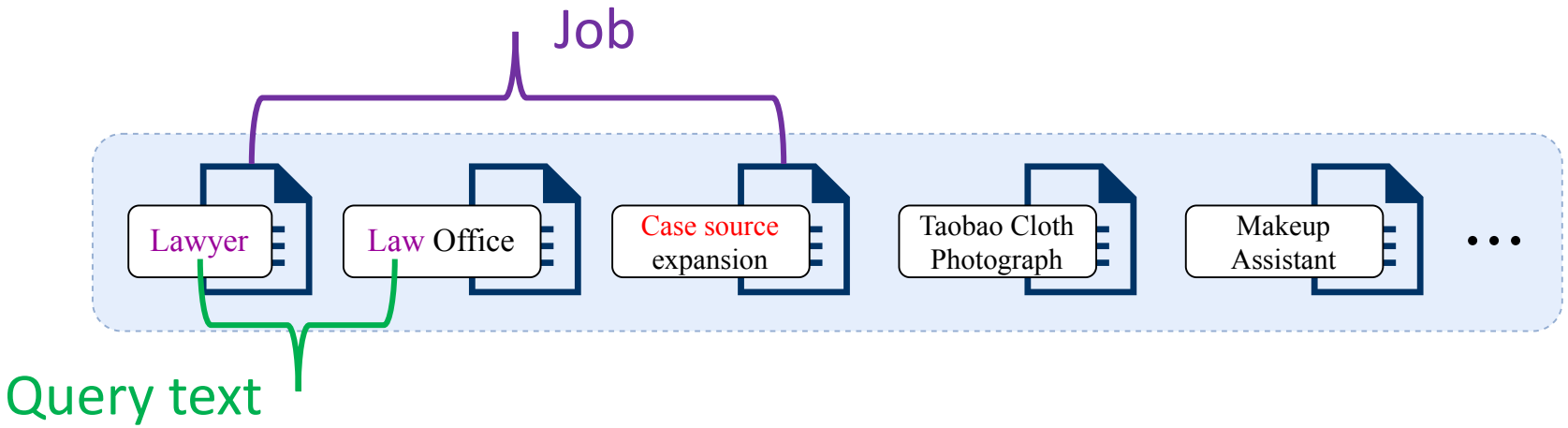
- Hard to capture job needs from search history



  - Data format: list of short queries together with clicked jobs;
  - Redundant & Diverse: how to find out underlying major intentions;

Idea

- 1. Cluster job intentions;
- 2. Assign attention weight on intentions based on:
  - Job similarity;
  - Query text similarity;



Problem Definition

- $u, j, \mathcal{H}_u \rightarrow y$

- Candidate: u
- Job: j
- Search history: $\mathcal{H}_u = \{\langle q_1, j_1 \rangle, \ldots, \langle q_L, j_L \rangle\}$

  <Query text, job ID>

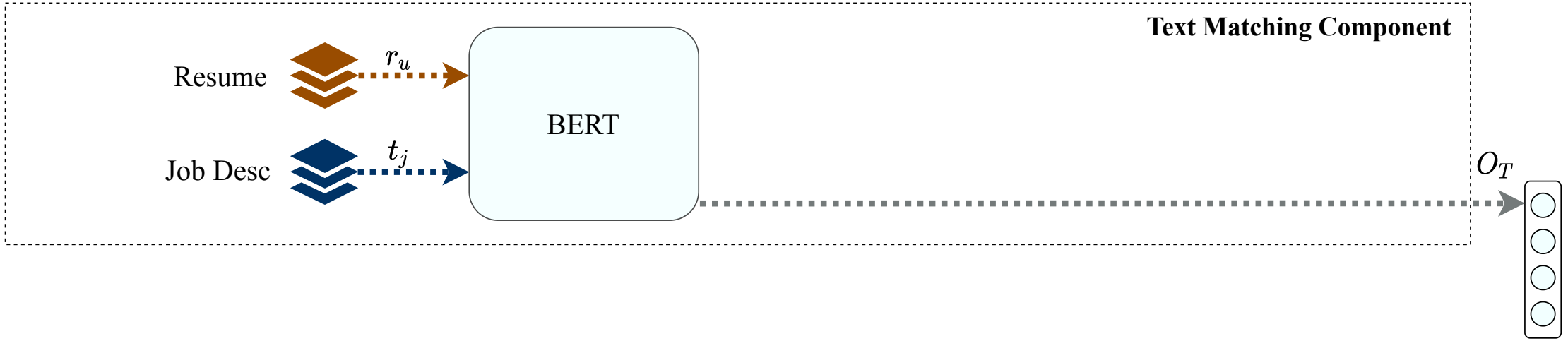
- Matching or not: y

Solution- SHPJF

- Search History enhanced Person-Job Fit

- Text Matching Component
- Intention Modeling Component

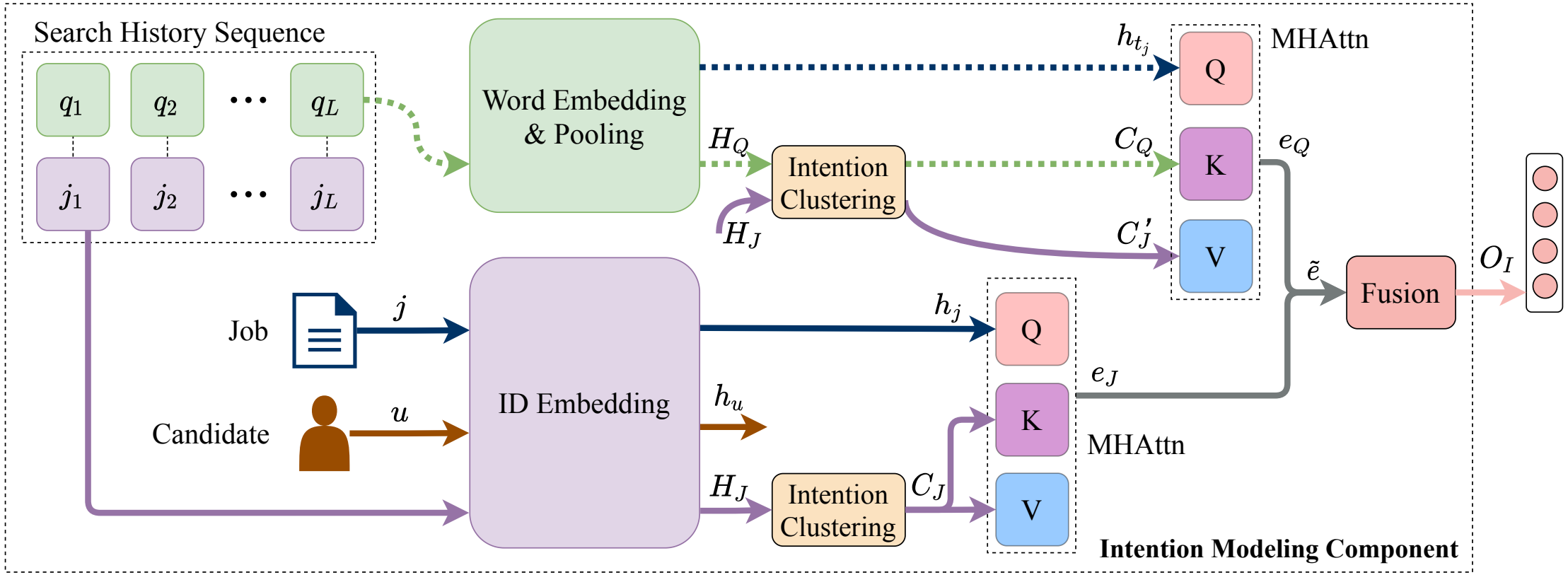
Solution – Text Matching



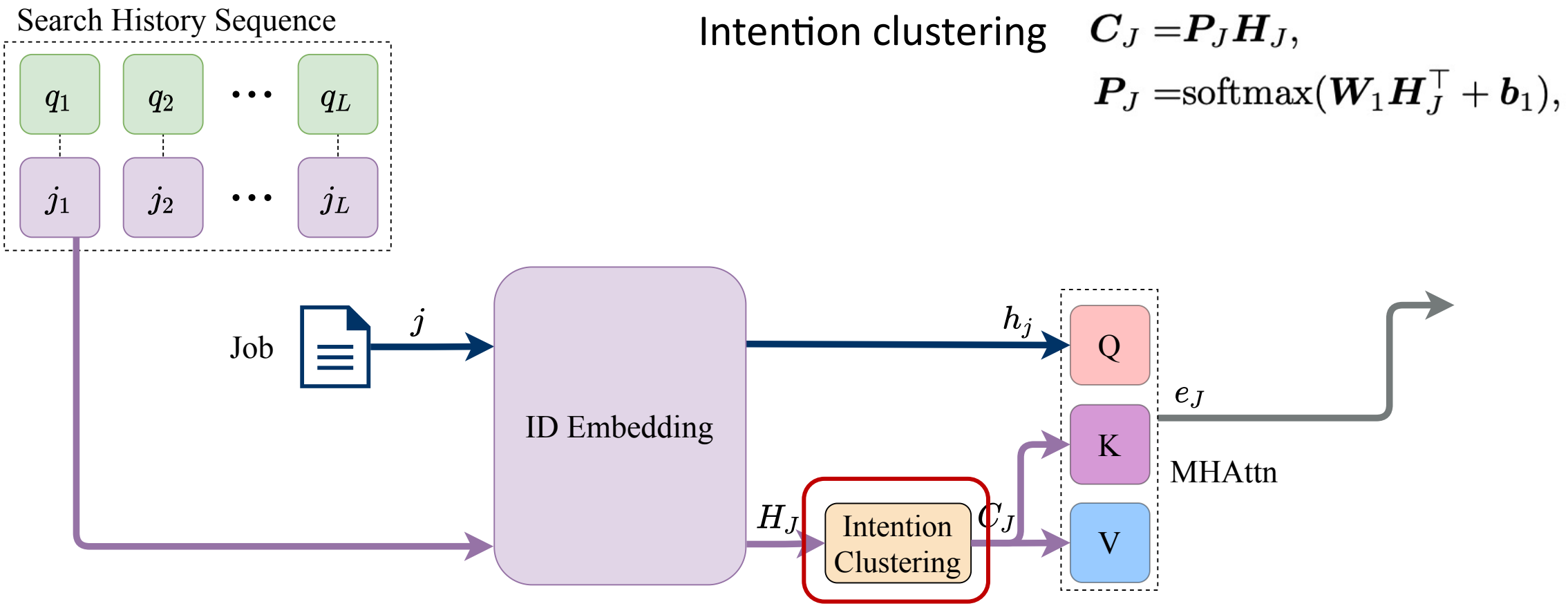
$$\boldsymbol{o}_T = \mathrm{BERT}([CLS]; r_u; [SEP]; t_j]),$$

single-tower encoder

Solution – Intention Modeling



Solution – Intention Modeling (Job)



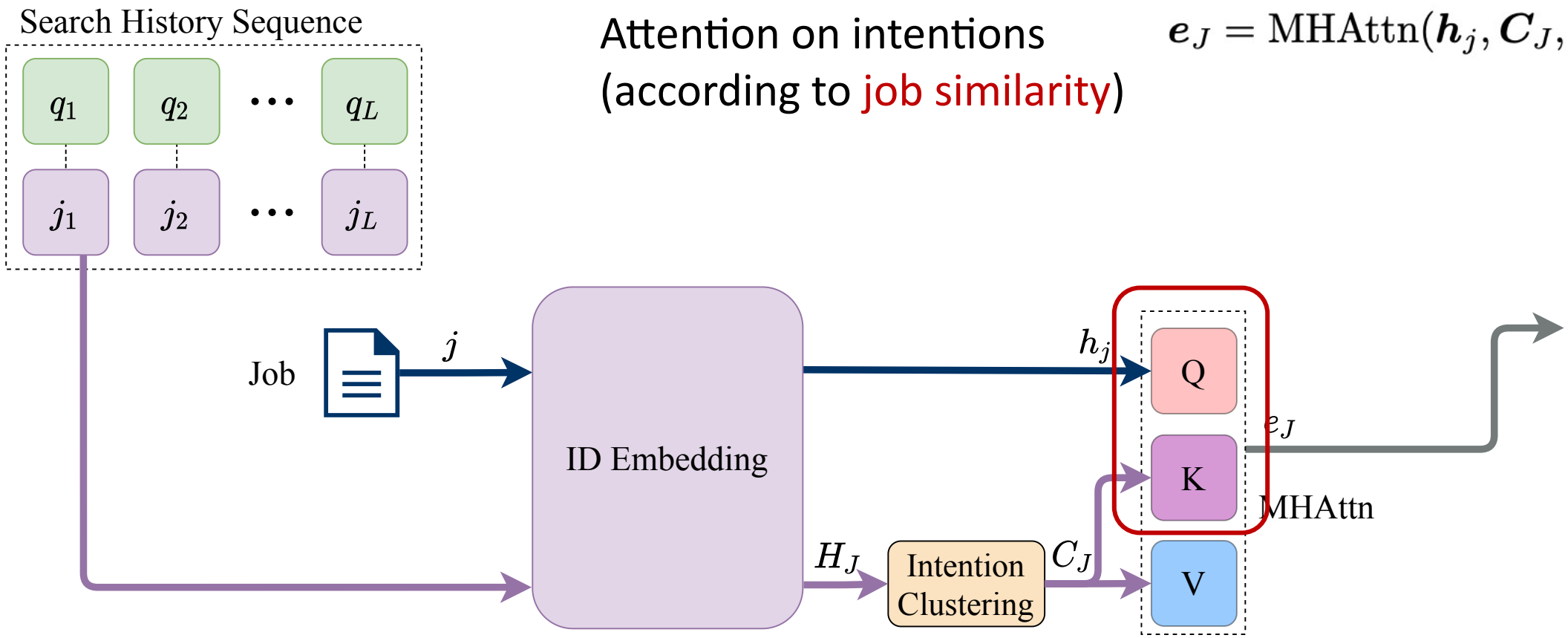
Intention clustering $\boldsymbol{C}_J = \boldsymbol{P}_J \boldsymbol{H}_J,$

$\boldsymbol{P}_J = \text{softmax}(\boldsymbol{W}_1 \boldsymbol{H}_J^\top + \boldsymbol{b}_1),$

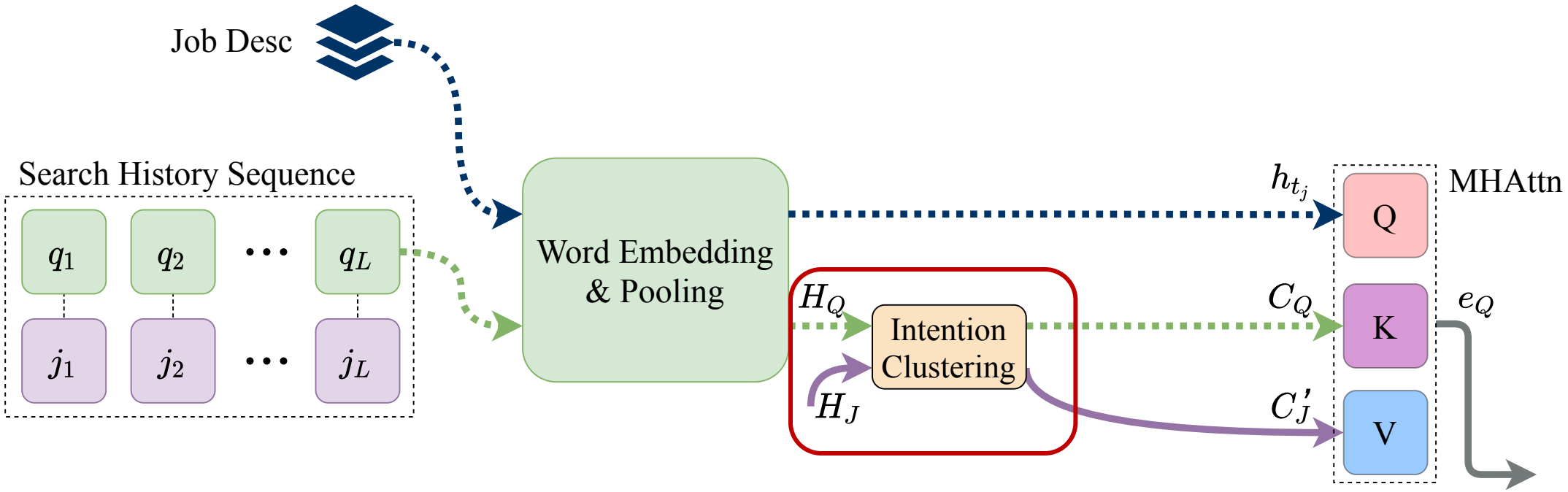
Solution – Intention Modeling (Job)

Attention on intentions
(according to job similarity)

$$\boldsymbol{e}_J = \mathrm{MHAttn}(\boldsymbol{h}_j, \boldsymbol{C}_J, \boldsymbol{C}_J),$$



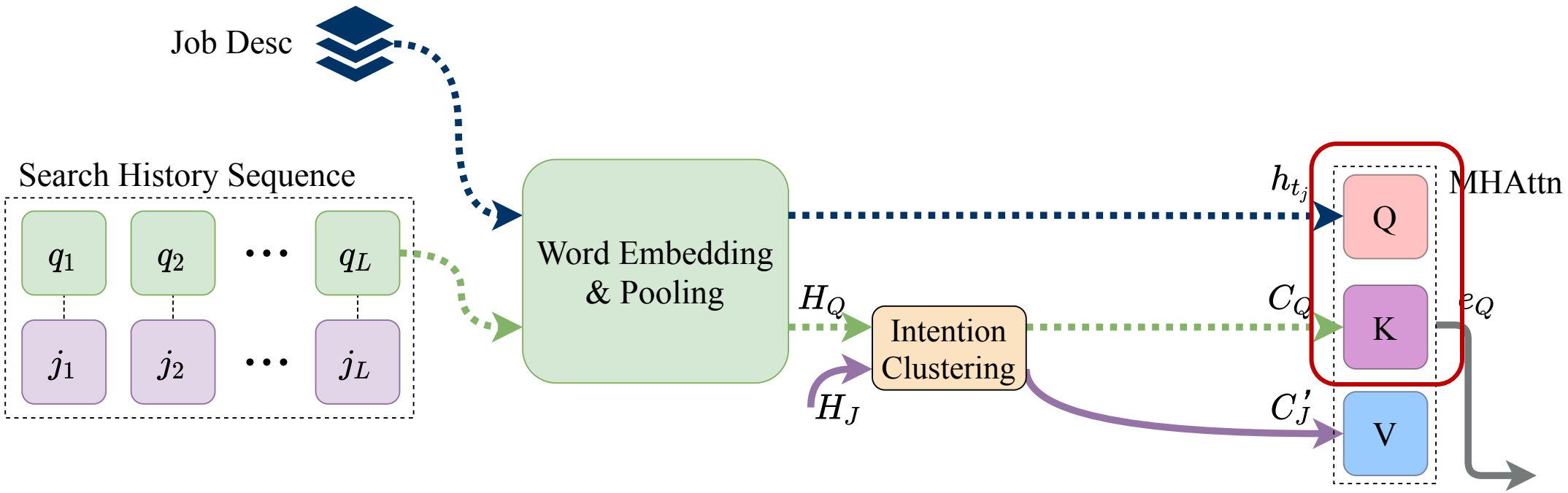
Solution – Intention Modeling (Text)



Intention clustering

$$\boldsymbol{C}_Q = \boldsymbol{P}_Q \boldsymbol{H}_Q,$$
$$\boldsymbol{C}'_J = \boldsymbol{P}_Q \boldsymbol{H}_J,$$
$$\boldsymbol{P}_Q = \text{softmax}(\boldsymbol{W}_2[\boldsymbol{H}_Q; \boldsymbol{H}_J]^\top + \boldsymbol{b}_2),$$

Solution – Intention Modeling (Text)



Attention on intentions
(according to text similarity)

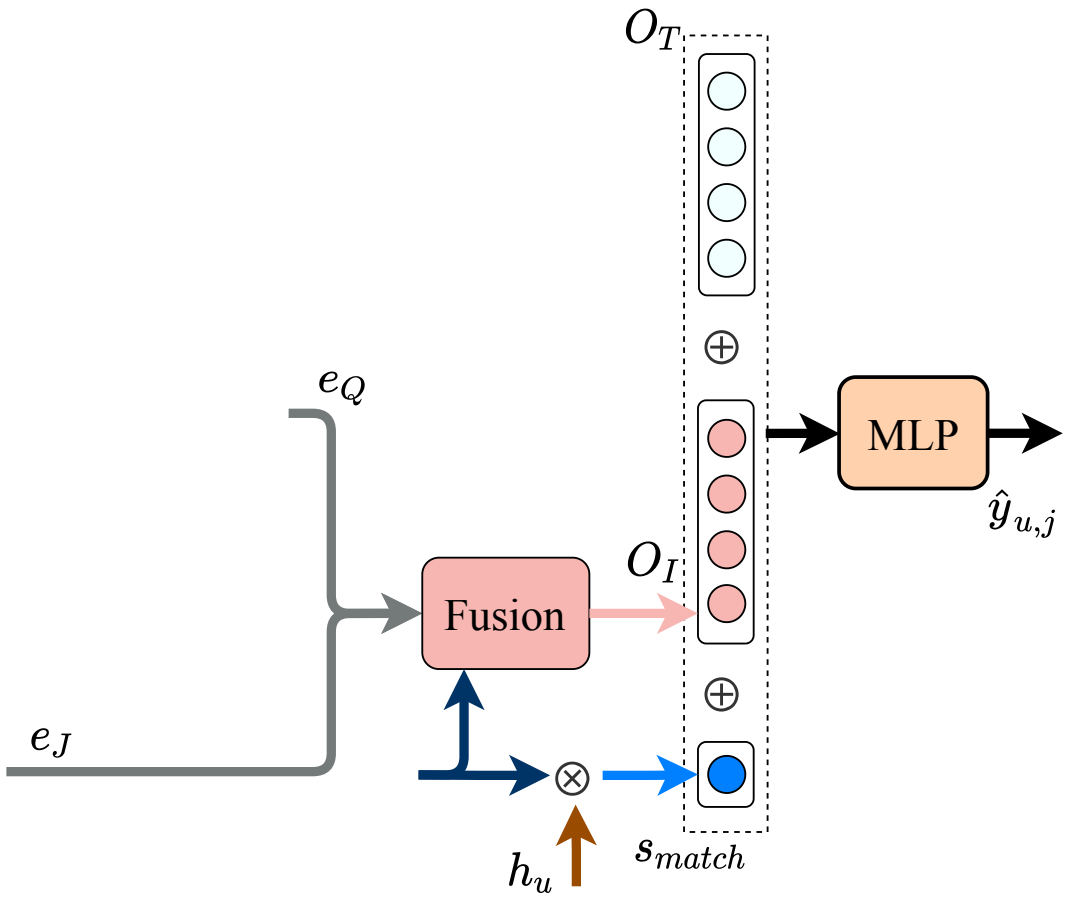
$$\boldsymbol{e}_Q = \mathrm{MHAttn}(\boldsymbol{h}_{t_j}, \boldsymbol{C}_Q, \boldsymbol{C}_J),$$

Solution – Fusion

$$\tilde{\boldsymbol{e}} = \lambda \boldsymbol{e}_J + (1-\lambda)\boldsymbol{e}_Q,$$

$$\boldsymbol{o}_I = \mathrm{MLP}\big([\tilde{\boldsymbol{e}}; \boldsymbol{h}_j; \tilde{\boldsymbol{e}} - \boldsymbol{h}_j; \tilde{\boldsymbol{e}} \circ \boldsymbol{h}_j]\big),$$

$$\hat{y}_{u,j} = \sigma(\mathrm{MLP}\big([\boldsymbol{o}_T; \boldsymbol{o}_I; s_{match}]\big)),$$



Experiments

- Dataset
  - Collect from BOSS Zhipin's real online log

#candidates	#jobs	#positive	#negative	$\overline{L}$	$\overline{\lvert q \rvert}$
53,566	307,738	257,922	2,109,876	16.55	1.50

Experiments

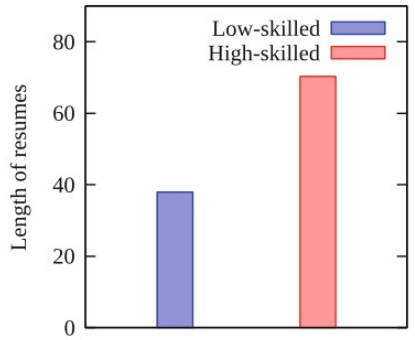
- Main results

Method	GAUC	R@1	R@5	MRR
PJFNN	0.5313	0.1412	0.5192	0.4025
BPJFNN	0.5343	0.1391	0.5217	0.4008
APJFNN	0.5323	0.1403	0.5185	0.4000
BERT	0.5449	0.1515	0.5297	0.4129
MV-CoN	0.5463	0.1554	0.5307	0.4165
SHPJF (ours)	**0.5785**	**0.1630**	**0.5516**	**0.4297**

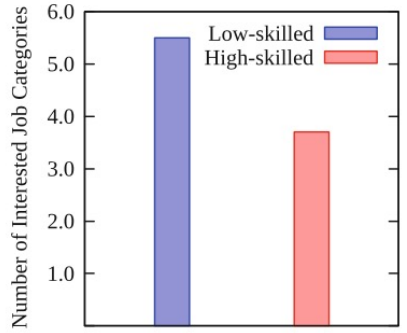
Experiments

- Evaluation in different skill groups
  - Short resumes
  - Multiple intentions

Groups	Low-skilled Candidates		High-skilled Candidates	
Method	GAUC	RelaImpr	GAUC	RelaImpr
PJFNN	0.5295	−4.53%	0.5318	−2.75%
BPJFNN	0.5399	+29.13%	0.5326	−0.31%
APJFNN	0.5309	0.00%	0.5327	0.00%
BERT	0.5381	+23.30%	0.5470	+43.73%
MV-CoN	0.5396	+28.16%	0.5484	+48.01%
SHPJF	**0.5689**	**+122.98%**	**0.5814**	**+148.93%**



(a) Average resume length.



(b) Average #interested categories.

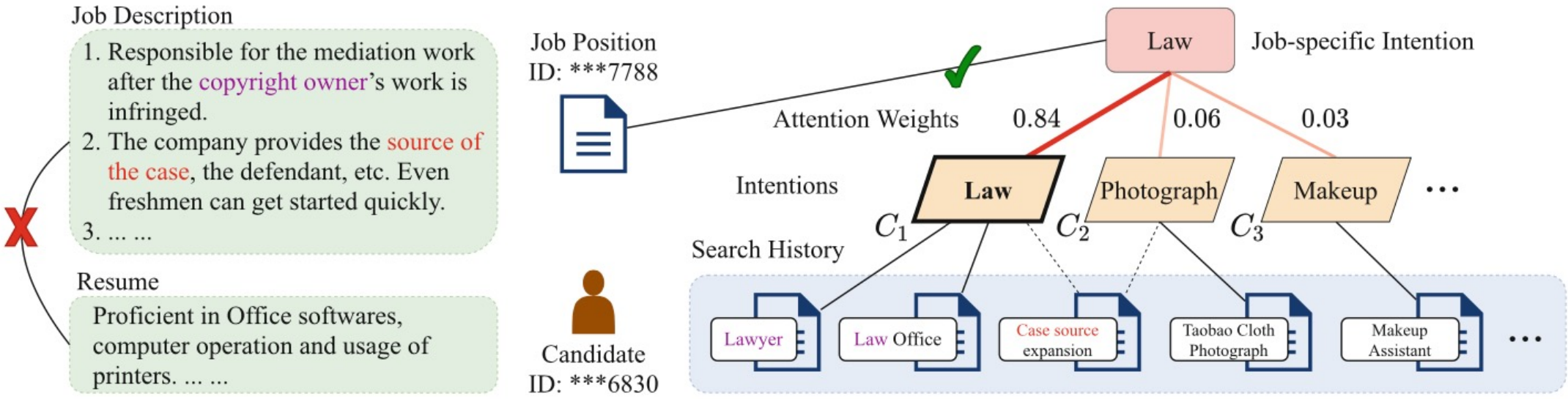
Experiments

- Ablation Study

Variants	GAUC	R@1	R@5	MRR
BERT	0.5449	0.1515	0.5297	0.4129
BERT_{GRU}	0.5557	0.1546	0.5334	0.4196
$\text{BERT}_{\text{query}}$	0.5572	0.1558	0.5342	0.4201
SHPJF w/o Q	0.5697	0.1599	0.5456	0.4270
SHPJF w/o J	0.5715	**0.1634**	0.5456	0.4286
SHPJF w/o C	0.5738	0.1581	0.5443	0.4237
SHPJF	**0.5785**	0.1630	**0.5516**	**0.4297**

Experiments

- Case Study



Conclusion

- SHPJF: Search History enhanced Person-Job Fit
  - Incorporating text/interactions from two channels: recommendation & search
  - Assign attention weight on intention representations
    - Text similarity
    - Job similarity
- Future Work
  - Side information from both channels

- https://github.com/RUCAIBox/SHPJF
- Welcome stars! 🌟