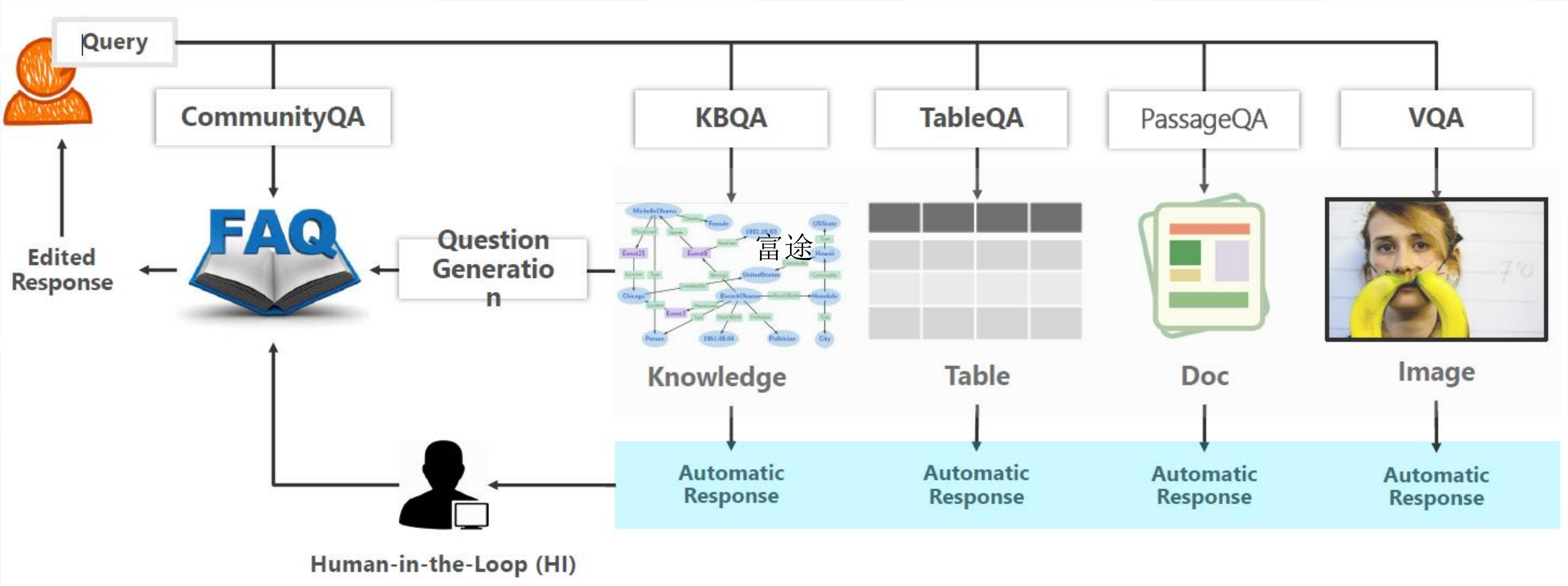
End-to-End quantum language model with Application to Question Answering

Peng Zhang[1], JiaBin Niu[1], Zhan Su[1], **Bengyou Wang**[2], Liqun Ma[1], Dawei Song[1]
Tianjin University[1]
Tencent[2]

Contents

QA System

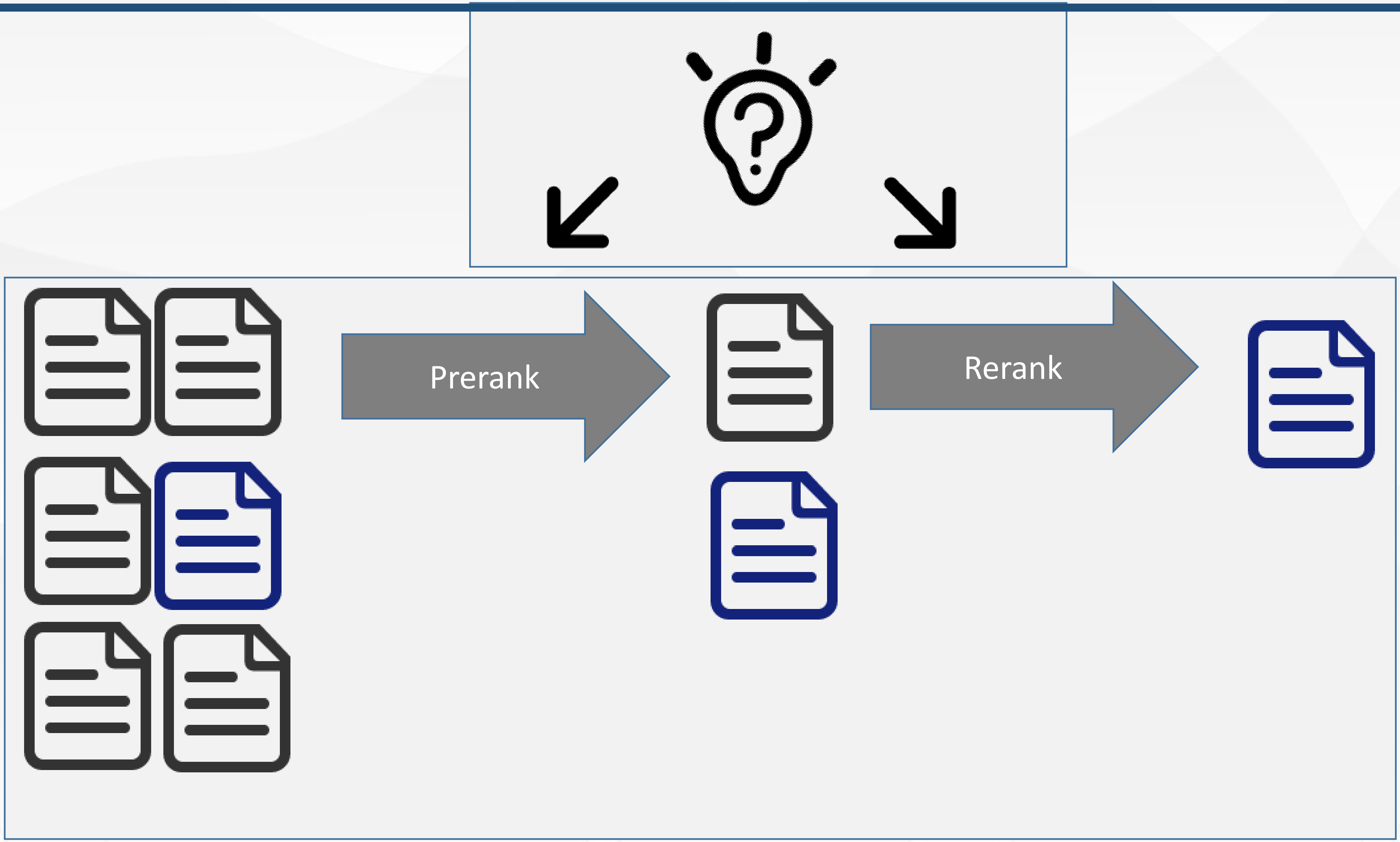


Query
CommunityQA
KBQA
TableQA
PassageQA
VQA
FAQ
Edited Response
Question Generatio n
富途
Knowledge
Table
Doc
Image
Automatic Response
Automatic Response
Automatic Response
Automatic Response
Human-in-the-Loop (HI)

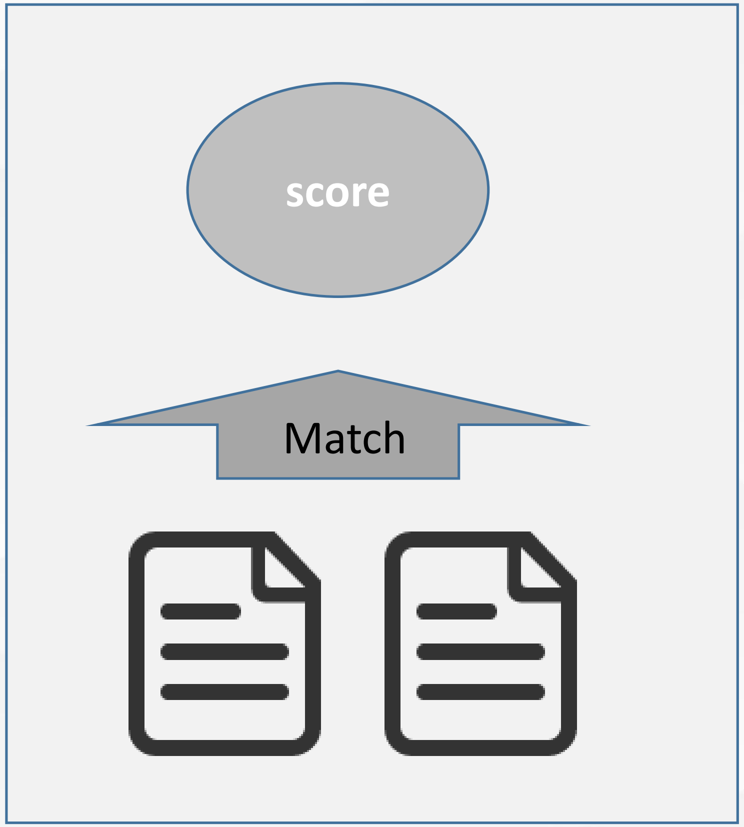
QA system in Tencent

- **Community QA**
  - FAQs
- KBQA
  - Knowledge Base
- Passage QA
  - Only unstructured documents

Two-step Architecture in Community QA



Textual Matching



- ✓ Unsupervised Models
  - ✓ TFIDF/BM25
  - ✓ language model
- ✓ Neural Network Models
  - ✓ DSSM
  - ✓ CNN/RNN variants

Contents

Statistical Language Model

- For a sequence of terms in the document $d=w_1w_2\ldots w_n$, SLM calculates the probability $P(w_1w_2\ldots w_n)$. Based on Beyes' rule, we have:

$$p(w_1w_2\cdots w_n)=p(w_1)p(w_2\cdots w_n\mid w_1)$$

$$=p(w_1)\prod_{i=2}^{n}p(w_i\mid w_{i-1}\cdots w_1)$$

SLM-based IR model (SLMIR)

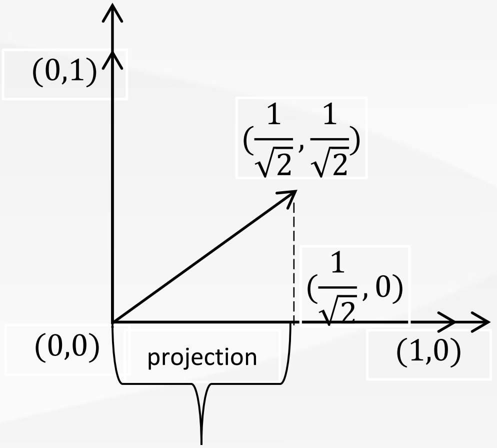
- **Query likelihood model:** define the relevance as the generative probability of the current query w.r.t. each document.
- **Translation model**: define the relevance as the probability that the query would have been generated as a translation of the document, and factor in the user's general preferences in the form of a prior distribution over documents.
- **KL-divergence model**： query and document are correspond to two different languages. And the relevance is defined as the KL-divergence between the two language models.

- **We focus on KL-divergence model in this talk.**

Contents

Quantum Concept

- Simple Example:



$|e_1\rangle = (1,0)^\top, |e_2\rangle = (0,1)^\top$

$s_1 = |e_1\rangle\langle e_1| = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$

$s_2 = |e_2\rangle\langle e_2| = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$

$s_3 = \frac{1}{\sqrt{2}}(s_1+s_2)$

$\vec{v} = (\frac{1}{\sqrt{2}},\frac{1}{\sqrt{2}})^{\mathrm{T}}$

$\text{Projection\_1} = s_1 \cdot \vec{v} = (\frac{1}{\sqrt{2}},0)^{\mathrm{T}}$

$\text{Projection\_2} = s_2 \cdot \vec{v} = (0,\frac{1}{\sqrt{2}})^{\mathrm{T}}$

$\text{Projection\_3} = s_3 \cdot \vec{v} = (\frac{1}{\sqrt{2}},\frac{1}{\sqrt{2}})^{\mathrm{T}}$

- ✓ A unit vector $\vec{u}\epsilon\mathbb{R}^n$, $\|\vec{u}\|_2 = 1$ is written as $|u\rangle$ (ket)
- ✓ The transpose $\vec{u}^\top$ is written as $\langle u|$ (*bra*)
- ✓ The projector onto the direction $\vec{u}$ writes as $|u\rangle\langle u|$ (*dyad*), corresponding to the pure state
- ✓ The inner product between two vectors writes as $\langle u|u\rangle$
- ✓ The elements of the standard basis in $\mathbb{R}^n$ are denoted as $|e_i\rangle = (\delta_{1i},\ldots,\delta_{ni})^\top, where\ \delta_{ij} = 1$, iff $i = j$
- ✓ Generally, any *ket* $|v\rangle = \sum_i v_i\,|u_i\rangle$ is called a *superposition* of the $|u_i\rangle$, where $\{|u_1\rangle,\ldots,|u_n\rangle\}$ form an orthonormal basis

Density Matrix

- Density Matrix

A density matrix corresponds to the discrete probability distribution in classical probability theory. It assigns a quantum probability to each one of the infinite dyads (an elementary event in quantum probability). For a density matrix ρ.

$$\rho = \begin{bmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{bmatrix}$$

$$\mu_\rho(|e\rangle\langle e|) \; = tr(\rho|e\rangle\langle e|) \; = 0.5, \; \mu_\rho(|f\rangle\langle f|) = tr(\rho|f\rangle\langle f|) = 1$$

where:

$$|e\rangle = \; (1,0)^\intercal \qquad |e\rangle\langle e| = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$$

$$|f\rangle = \; (\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}})^{\mathrm{T}} \qquad |f\rangle\langle f| = \begin{bmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{bmatrix}$$

Gleason's Theorem : A. Gleason. Measures on the closed subspaces of a hilbert space. Journ. Math. Mech., 6:885–893, 1957.

Quantum Language Models (QLM)

- *For example*:

V = *{computer, architecture, system}*, W_d=*{computer, architecture}*

◆If we only observe single words:

$$\mathcal{P}_d = \{\mathcal{E}_{computer}, \mathcal{E}_{architecture}\}$$

$$\mathcal{E}_{computer} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \mathcal{E}_{computer} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

◆If we observe the dependency of "*computer*" *and* "*architecture*"

$$k_{ca} = \sigma_c|e_c\rangle + \sigma_a|e_a\rangle, \text{Set } \sigma_c = \sqrt{2/3}, \sigma_\mathrm{a} = \sqrt{1/3}$$

$$\mathcal{K}_{ca}=\begin{bmatrix} \frac{2}{3} & \frac{\sqrt{2}}{3} & 0 \\ \frac{\sqrt{2}}{3} & \frac{2}{3} & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

N-gram in extended Vector Space

Word/Term Dependency	Computer	*Architecture*	*System*	Computer *Architecture*	Computer *System*	*Architecture System*	Computer *System Architecture*
Count	10	6	5	4	3	2	0
Frequency	0.33	0.2	0.166	0.133	0.1	0.066	0

$$P(W \mid \theta_d) = [0.33,\ 0.2,\ 0.166,\ 0.133,\ 0.1,\ 0.066,\ 0]$$

$$|V| = C_n^1 + C_n^2 + \cdots + C_n^n = \sum_{i=0}^{n} C_n^i$$

The dimension of parameter in extended Vector Space : **o(n!)**

Term Dependency (N-gram) in QLM

Word/Term Dependency	Computer	*Architecture*	*System*	Computer *Architecture*	Computer *System*	*Architecture* *System*	Computer *System* *Architecture*
Projection	e_c=[1,0,0]	e_a=[0,1,0]	e_s=[0,0,1]	$k_{ac}=[\frac{\sqrt{2}}{2},\frac{\sqrt{2}}{2},0]$	$k_{cs}=[\frac{\sqrt{2}}{2},0,\frac{\sqrt{2}}{2}]$	$k_{as}=[0,\frac{\sqrt{2}}{2},\frac{\sqrt{2}}{2}]$	$k_{cas}=[\frac{\sqrt{3}}{3},\frac{\sqrt{3}}{3},\frac{\sqrt{3}}{3}]$
Frequency	$\mathrm{Tr}(\rho\|e_c\rangle\langle e_c\|)$	$\mathrm{Tr}(\rho\|e_a\rangle\langle e_a\|)$	$\mathrm{Tr}(\rho\|e_s\rangle\langle e_s\|)$	$\mathrm{Tr}(\rho\|k_{ac}\rangle\langle k_{ac}\|)$	$\mathrm{Tr}(\rho\|k_{cs}\rangle\langle k_{cs}\|)$	$\mathrm{Tr}(\rho\|k_{as}\rangle\langle k_{as}\|)$	$\mathrm{Tr}(\rho\|k_{cas}\rangle\langle k_{cas}\|)$

$$\rho = \begin{bmatrix} \frac{2}{3} & \frac{\sqrt{2}}{3} & 0 \\ \frac{\sqrt{2}}{3} & \frac{2}{3} & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$P(W|\theta_d) = [\mathrm{Tr}(\rho|e_c\rangle\langle e_c|), \mathrm{Tr}(\rho|e_a\rangle\langle e_a|), \mathrm{Tr}(\rho|e_s\rangle\langle e_s|), \mathrm{Tr}(\rho|k_{ac}\rangle\langle k_{ac}|), \mathrm{Tr}(\rho|k_{cs}\rangle\langle k_{cs}|), \mathrm{Tr}(\rho|k_{as}\rangle\langle k_{as}|), \mathrm{Tr}(\rho|k_{cas}\rangle\langle k_{cas}|)]$$

The dimension of parameter in QLM: **o(n^2)**

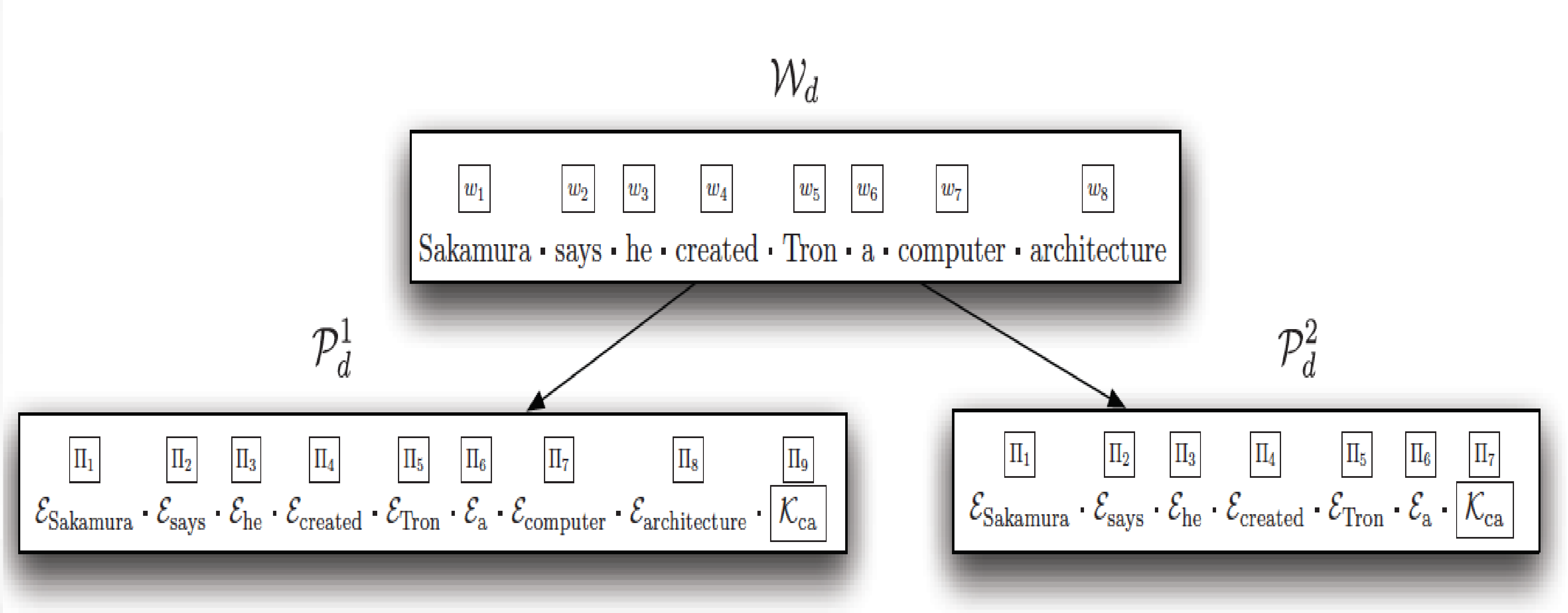
Term Dependency in Quantum Entanglement

Definition 1. (QE): Let A be an n-qubit system in a state $|\phi_A\rangle$ and $\{A_1, A_2\}$ be a partition of A, where two disjoint parts A_1 and A_2 have $0 < k < n$ qubits and $n - k$ qubits, respectively. A is entangled iff. there does NOT exist any tensor product decomposition of $|\phi_A\rangle$ such that $|\phi_A\rangle = |\phi_{A_1}\rangle \otimes |\phi_{A_2}\rangle$, where $|\phi_{A_1}\rangle$ and $|\phi_{A_2}\rangle$ are the states of A_1 and A_2, respectively.

Definition 2. (UPD): A pattern $A = \{W_1, W_2, \cdots, W_n\}$ forms the UPD pattern iff. the joint probability distribution over A cannot be unconditionally factorized, i.e., there does NOT exist any m-partition $\{A_1, A_2, \cdots, A_m; m > 1\}$ of A, so that $p(\boldsymbol{a}) = p(\boldsymbol{a}_1) \cdot p(\boldsymbol{a}_2) \cdots p(\boldsymbol{a}_m)$, where $p(\boldsymbol{a}_i)$, $i = 1, 2, \cdots, m$, is the joint distribution over A_i.

- Modeling quantum entanglements in quantum language models. **IJCAI 2015**

Quantum Language Model (QLM)



- LM: a document *d* is represented by a sequence of terms
- QLM: *d* is represented by a sequence of quantum events (with dyads for a term or a dependency)

[Sordoni, Nie, Bengio , 2013]

Computing probabilities

$$p(\mathrm{s}) = \sum p(\varphi_i)\, |\langle s|\varphi_i\rangle|^2$$

$$= \sum p(\varphi_i)\, \langle \varphi_i|\hat{M}_s{}^{\dagger}\hat{M}_s|\varphi_i\rangle$$

$$= \sum p(\varphi_i)\, tr\big(\hat{M}_s|\varphi_i\rangle\langle\varphi_i|\big)$$

$$= \mathrm{tr}\left(\hat{M}_s \underbrace{p(\varphi_i)\sum|\varphi_i\rangle\langle\varphi_i|}_{\rho}\right)$$

$$= \mathrm{tr}(\hat{M}_s\rho)$$

$$= \mathrm{tr}(\rho\hat{M}_s)$$

Where $\boldsymbol{\rho}$ is a density matrix

Measurement in QLM

- A density matrix ρ to represent sentence
- Given the observed projectors $\mathcal{P}_d = \{\Pi_1, \dots, \Pi_M\}$ for sentence S, the quantum language model ρ is estimated through Maximum Likelihood Estimation, and the likelihood is represented as：

- $\mathcal{L}_{\mathcal{P}_d}(\rho) = \prod_{i=1}^{M} tr(\rho \Pi_i)$

Maximum likelihood estimation for QLM

- Likelihood:

$$\mathcal{L}_{\mathcal{P}_d}(\rho) = \prod_{i=1}^{M} \operatorname{tr}(\rho \Pi_i).$$

- Estimation/Training of Density Matrix:

$$\begin{array}{ll} \underset{\rho}{\text{maximize}} & \log \mathcal{L}_{\mathcal{P}_d}(\rho) \\ \text{subject to} & \rho \in \mathcal{S}^n. \end{array}$$

- Matching:

$$\begin{aligned} -\Delta_{VN}(\rho_q \| \rho_d) &= -\operatorname{tr}(\rho_q(\log \rho_q - \log \rho_d)) \\ &\overset{rank}{=} \operatorname{tr}(\rho_q \log \rho_d), \end{aligned}$$

Limitation in QLM

- **If the two documents do not share any words, especially in short text**
  - *Use embedding as a basic vector*

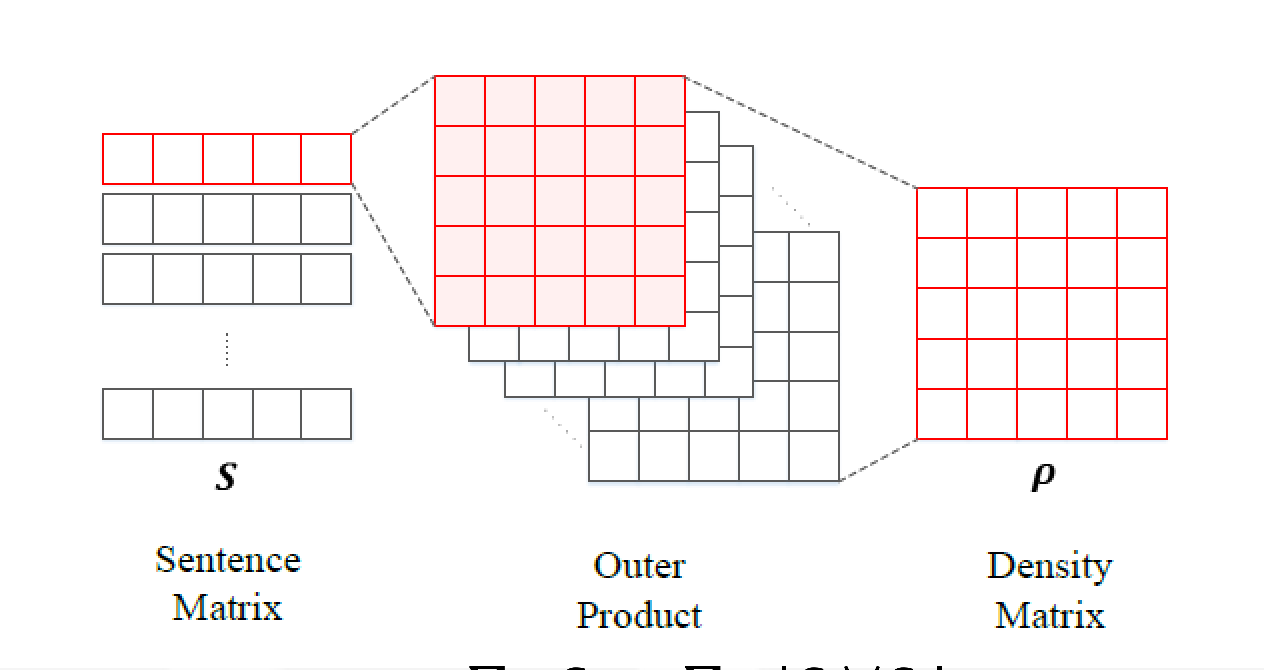
- **It is independent with the label。**
  - *Training in a end-2-end network*

Neural Network based Quantum Language Model

Contents

Neural Network based QLM (NNQLM)

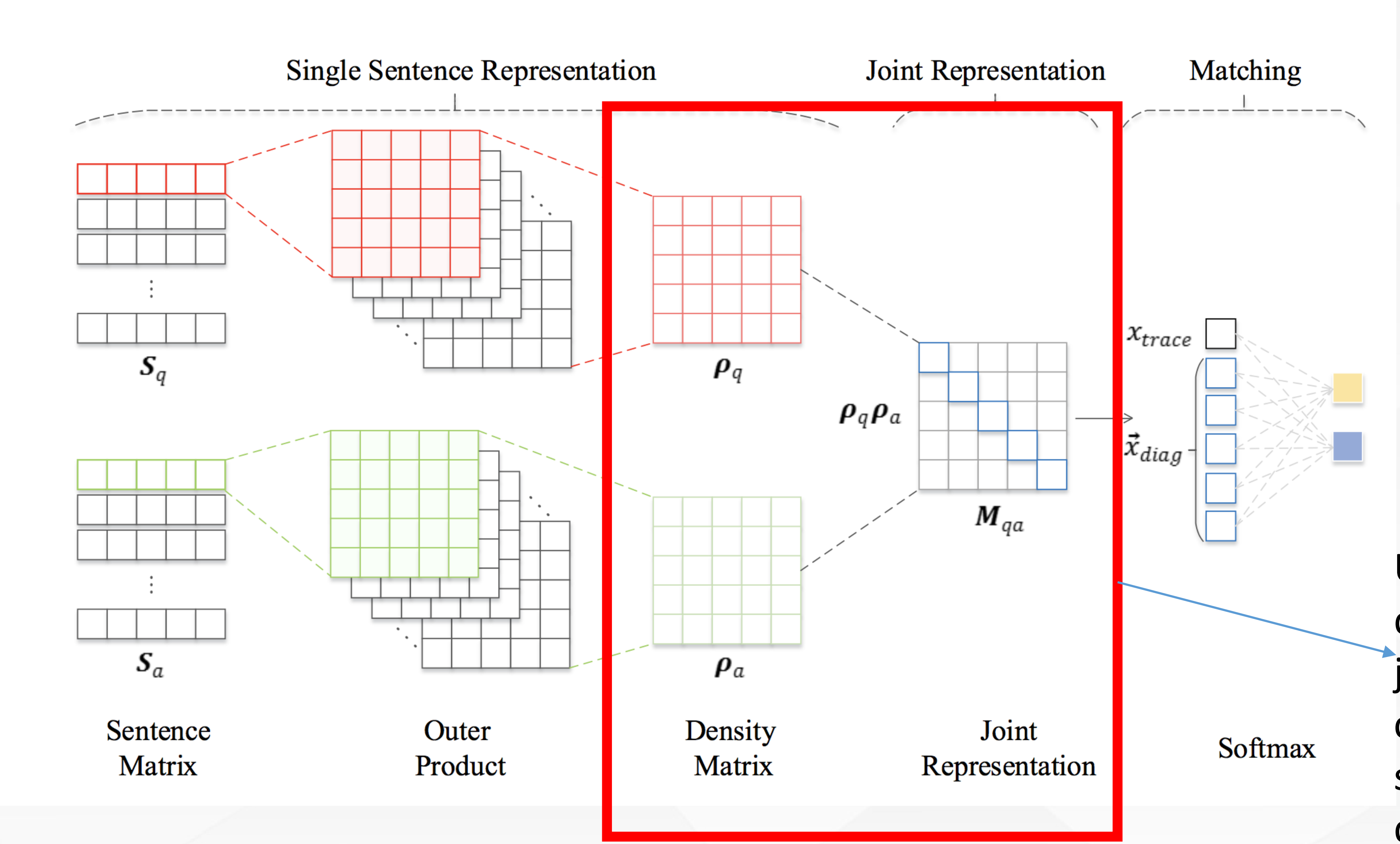
- Density matrix representation for sentences (q or a)



$$\rho = \sum p_i S_i = \sum p_i |S_i\rangle\langle S_i|$$

$$|S_i\rangle = \frac{\vec{S_i}}{\left|\left|\vec{S_i}\right|\right|}$$

A relatively simple architecture (NNQLM-I)



Using the product of the density matrixes as their joint representation, . The combined representations show the similarity of their density matrices.

inter-sentence similarities

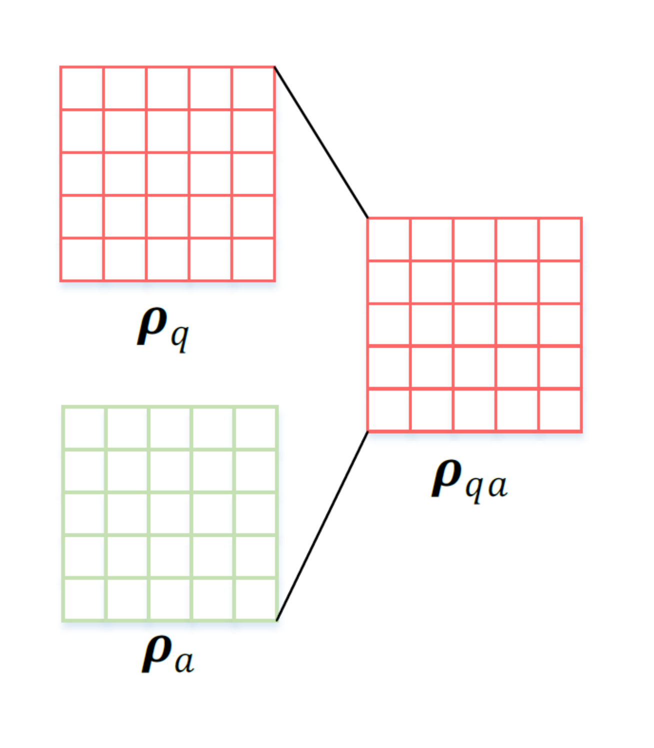
- Since the density matrix is semi-positive, it

$$\rho_q = \sum_i \lambda_i |r_i\rangle\langle r_i|$$

$$\rho_a = \sum_j \lambda_j |r_j\rangle\langle r_j|$$

$$\begin{aligned}\rho_q\rho_a &= \sum_{i,j} \lambda_i\lambda_j |r_i\rangle\langle r_i|r_j\rangle\langle r_j| \\ &= \sum_{i,j} \lambda_i\lambda_j \langle r_i|r_j\rangle|r_i\rangle\langle r_j|\end{aligned}$$

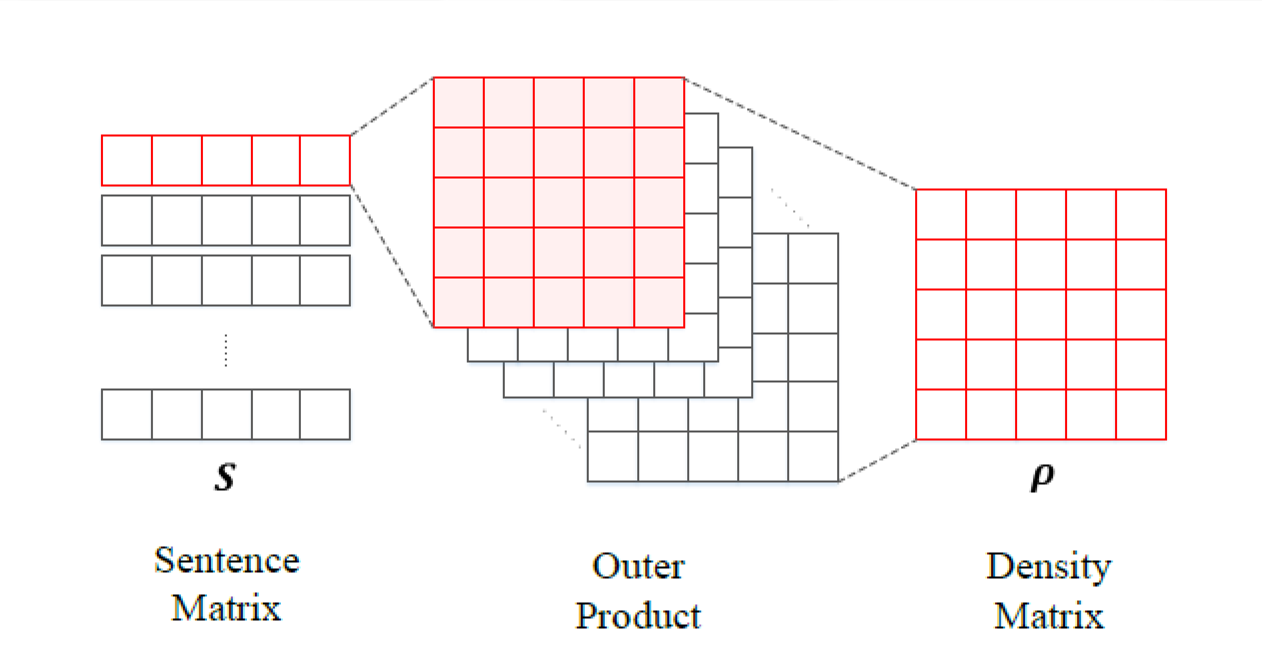
$$\mathrm{tr}(\rho_q\rho_a) = \sum_{i,j} \lambda_i\lambda_j \langle r_i|r_j\rangle^2$$



the similarity between ρ_q and ρ_a

Simple version: NNQLM1

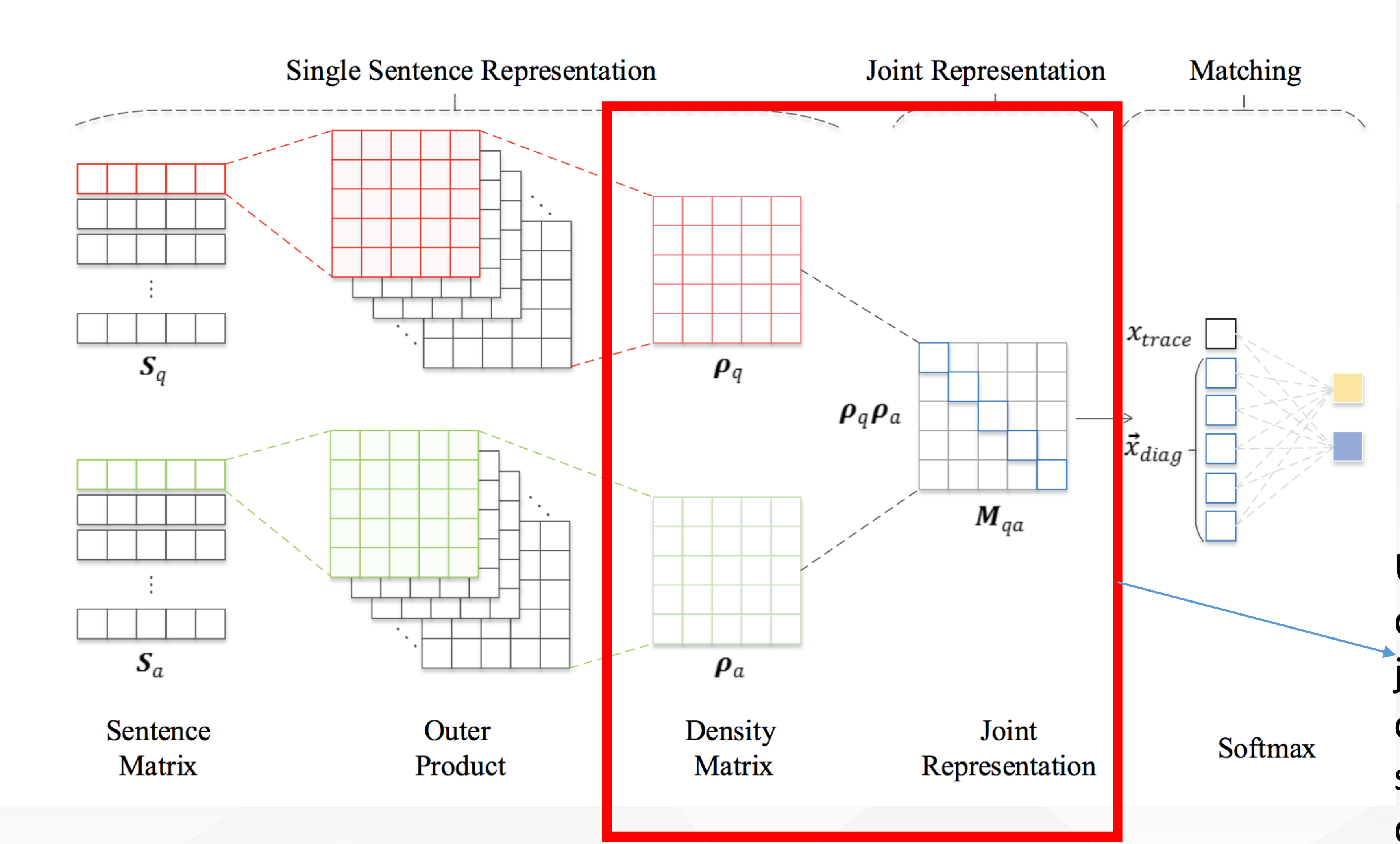
- Density matrix representation for sentences (q or a)



$$\rho = \sum p_i S_i = \sum p_i |S_i\rangle\langle S_i|$$

$$|S_i\rangle = \frac{\vec{S_i}}{\left|\left|\vec{S_i}\right|\right|}$$

Architecture of NNQLM1



Using the product of the density matrixes as their joint representation, . The combined representations show the similarity of their density matrices.

Inter-sentence Similarities

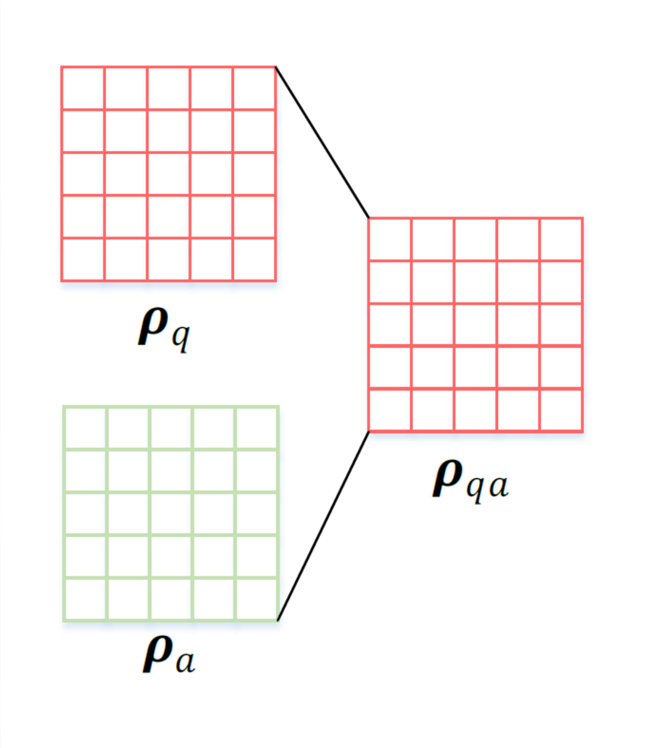
- Since the density matrix is semi-positive, it

$$\boldsymbol{\rho}_q = \sum_i \lambda_i \, |r_i\rangle\langle r_i|$$

$$\boldsymbol{\rho}_a = \sum\nolimits_j \lambda_j \, |r_j\rangle\langle r_j|$$

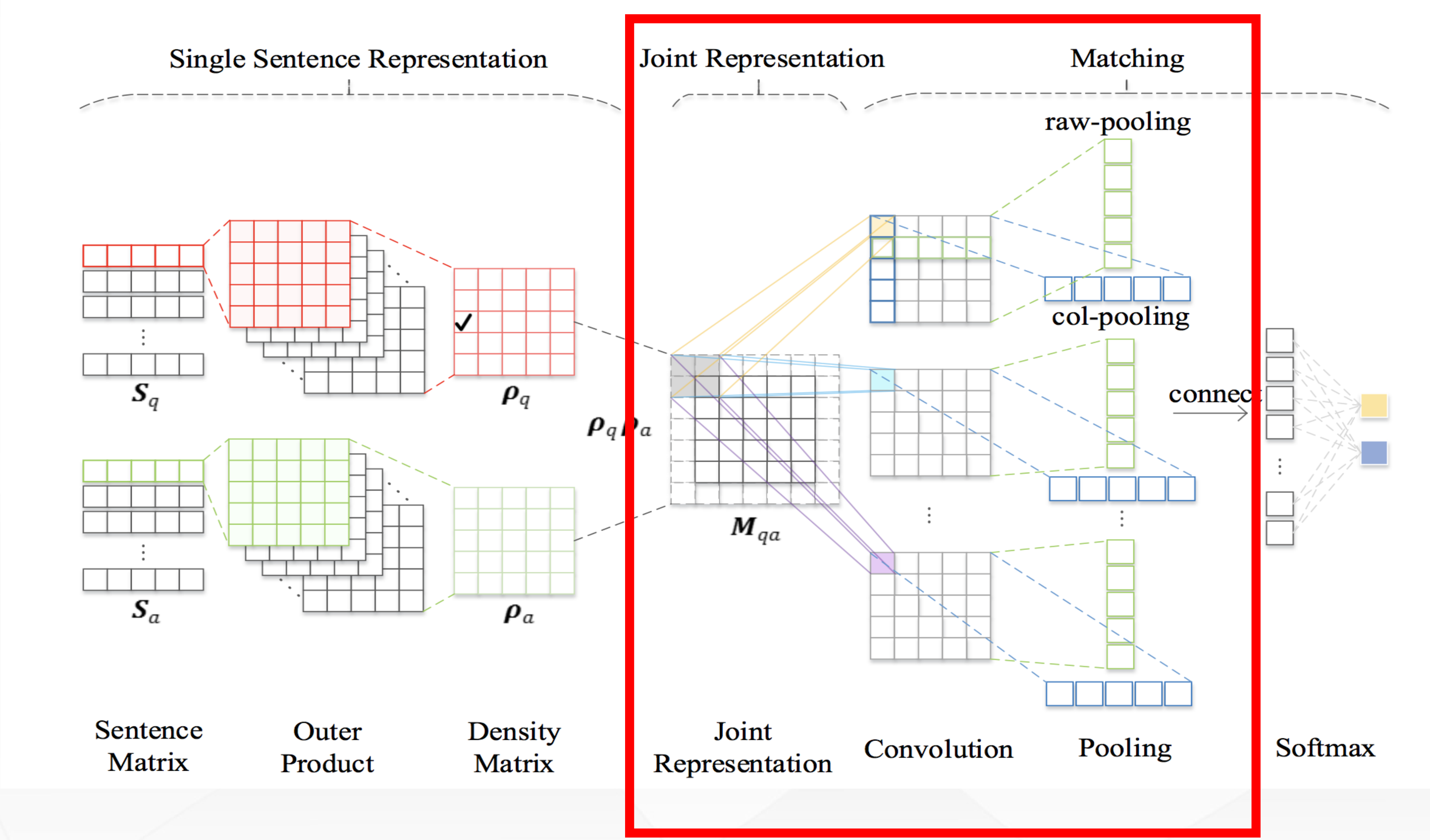
$$\begin{aligned} \boldsymbol{\rho}_q \boldsymbol{\rho}_a &= \sum_{i,j} \lambda_i \lambda_j \, |r_i\rangle\langle r_i|r_j\rangle\langle r_j| \\ &= \sum_{i,j} \lambda_i \lambda_j \, \langle r_i|r_j\rangle |r_i\rangle\langle r_j| \end{aligned}$$

$$\operatorname{tr}(\boldsymbol{\rho}_q \boldsymbol{\rho}_a) = \sum_{i,j} \lambda_i \lambda_j \, \langle r_i|r_j\rangle^2$$



the similarity between ρ_q and ρ_a

Architecture in NNQLM2



Future work in Quantum-inspired NN

- **Complex embedding**

  Richer input, higher performance

- **Interference in NN,**

  Cross-modal fusion

- **Entanglement** in NN

  Connection and memory in NN

More works try to bridge the gap between ***Quantum Concept and Deep learning*** *[1],*
It may open a new door to reveal the ***black-box inner mechanism*** *of Neural Network*

- [1] Deep Learning and Quantum Entanglement: Fundamental Connections with Implications to Network Design. **ICLR 2018**

Contents

- QA System
- Statistical Language Model
- Quantum Language Model
- NN-based Quantum Language Model
- **Quantum AI**

Exploration in Quantum AI

国务院印发《新一代人工智能发展规划》，明确提出
布局量子人工智能、自然语言处理、社会计算等领域

- Machine learning algorithm in Quantum computer

✓ **Quantum-inspired models and ideas, but not depends on Quantum Computer**

Quantum on general AI

- Solving the quantum many-body problem with artificial neural networks[J]. **Science**, 2017
- Deep Learning and Quantum Entanglement: Fundamental Connections with Implications to Network Design. **ICLR 2018**
- Deep complex Network. **ICLR 2018**
- Efficient representation of quantum many-body states with deep neural networks. **Nature Communications**. 2017
- SchNet: A continuous-filter convolutional neural network for modeling quantum interactions, **NIPS 2017**

Quantum AI on Language

- End-to-End Quantum-like Language Models with Application to Question Answering. **AAAI 2018**.
- Modeling multi-query retrieval tasks using density matrix transformation. **SIGIR 2015**
- Modeling quantum entanglements in quantum language models. **IJCAI 2015**
- Learning Concept Embeddings for Query Expansion by Quantum Entropy Minimization. **AAAI 2014**
- Modeling latent topic interactions using quantum interference for information retrieval. **CIKM 2013**
- Modeling term dependencies with quantum language models for IR. **SIGIR 2013**
- Pure high-order word dependence mining via information geometry, **ICTIR 2011 best paper**.
- A novel re-ranking approach inspired by quantum measurement. **ECIR 2011 best paper**.