Learning to Match
（匹配学习）

Hang Li
Noah’s Ark Lab
Huawei Technologies

* Work was done at Microsoft Research, with former colleagues and interns
The goal of life is to make your heartbeat match the beat of the universe, to match your nature with Nature
-- Joseph Campbell

生命的目的在于将自己的脉搏与宇宙的脉搏匹配起来，将自己的天赋与自然匹配起来
-- 约瑟夫·坎贝尔
Matching between Heterogeneous Objects is Everywhere

• Search: matching between queries and documents
• Recommendation: matching between users and products
• Image annotation: matching between keywords and images
• Drug design: matching between receptors and ligands
• Matchmaking: matching between men and women
• ... ...
Example: Image Annotation

- hook
- fishing
- singer
- solider
- worrier
- microphone
Talk Outline

• Learning to Match
• Learning to Match in Search
• Our Approaches to Learning to Match
  – Regularized Latent Semantic Indexing
  – Matching in Latent Space
  – String Re-Writing Kernel
• Summary
Learning to Match
(匹配学习)
Matching Problem: Instance Matching

<table>
<thead>
<tr>
<th></th>
<th>d1</th>
<th>d2</th>
<th>d3</th>
<th>dn</th>
</tr>
</thead>
<tbody>
<tr>
<td>q1</td>
<td>1</td>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>q2</td>
<td></td>
<td>4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>q3</td>
<td></td>
<td></td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>qm</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

IDs

Can also be represented as bi-partite graph
## Matching Problem: Content Matching

<table>
<thead>
<tr>
<th>Features</th>
<th>d1</th>
<th>d2</th>
<th>d3</th>
<th>dn</th>
</tr>
</thead>
<tbody>
<tr>
<td>q1</td>
<td></td>
<td></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>q2</td>
<td></td>
<td></td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>q3</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>qm</td>
<td>1</td>
<td></td>
<td>5</td>
<td></td>
</tr>
</tbody>
</table>

Can also be viewed as matching between two feature spaces.
Matching Problem: Content Matching

<table>
<thead>
<tr>
<th>Structures</th>
<th>d1</th>
<th>d2</th>
<th>d3</th>
<th>dn</th>
</tr>
</thead>
<tbody>
<tr>
<td>q1</td>
<td></td>
<td></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>q2</td>
<td></td>
<td></td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>q3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>qm</td>
<td></td>
<td></td>
<td>1</td>
<td>5</td>
</tr>
</tbody>
</table>
Formulation of Learning to Match

• Learning matching function
  \[ f(x, y) \]

• Training data \((x_1, y_1, r_1), \ldots, (x_N, y_N, r_N)\)

• Generated according to
  \[ x \sim P(X), \quad y \sim P(Y | X), \quad r \sim P(R | X, Y) \]

• Relations exist between \(x\) and \(y\)

• Two-input model differs from one-input model
Formulation of Learning to Match

- **Loss Function**
  \[ L(r, f(x, y)) \]

- **Risk Function**
  \[ R(r, f(x, y)) = \int_{X \times Y \times R} P(x, y, r) L(r, f(x, y)) dP(x, y, r) \]

- **Objective Function in Learning**
  \[ \min_{f \in F} \sum_{i=1}^{N} L(r_i, f(x_i, y_i)) + \Omega(f) \]
Challenges in Learning to Match

• Data is sparse
• Scale is very large
• Data structure is complicated
• Additional knowledge is available
Why to Coin the Term?

• ‘Learning to Match’ has been studied separately in different areas
• It is necessary to
  – Generalize the problems and solutions
  – Study the general methodology and theory
• Extensions can be considered, e.g.,
  – Multiple dimensions
  – Combination with graph or hyper-graph
Learning to Match in Search
Same Search Intent Different Query Representations
Example = “Distance between Sun and Earth”

- "how far" earth sun
- "how far" sun
- "how far" sun earth
- average distance earth sun
- average distance from earth to sun
- average distance from the earth to the sun
- distance between earth & sun
- distance between earth and sun
- distance between earth and the sun
- distance from earth to the sun
- distance from sun to earth
- distance from sun to the earth
- distance from the earth to the sun
- distance from the sun to earth
- distance of earth from sun
- distance between earth sun
- how far away is the sun from earth
- how far away is the sun from the earth
- how far earth from sun
- how far earth is from the sun
- how far earth is the sun
- how far from earth is the sun
- how far from earth to sun
- how far from earth to the sun
- distance between sun and earth
Same Search Intent, Different Query Representations

Example = “Youtube”

<table>
<thead>
<tr>
<th>Query</th>
<th>Simplified Query</th>
<th>Full Query</th>
</tr>
</thead>
<tbody>
<tr>
<td>yutube</td>
<td>yuotube</td>
<td>yuo tube</td>
</tr>
<tr>
<td>ytube</td>
<td>youtubr</td>
<td>yu tube</td>
</tr>
<tr>
<td>youtubo</td>
<td>youtuber</td>
<td>youtubecom</td>
</tr>
<tr>
<td>youtube om</td>
<td>youtube music videos</td>
<td>youtube videos</td>
</tr>
<tr>
<td>youtube</td>
<td>youtube com</td>
<td>youtube co</td>
</tr>
<tr>
<td>youtub com</td>
<td>you tube music videos</td>
<td>youtube</td>
</tr>
<tr>
<td>youtub</td>
<td>you tube com youtube</td>
<td>your tube</td>
</tr>
<tr>
<td>you tube</td>
<td>you tub</td>
<td>you tube video clips</td>
</tr>
<tr>
<td>you tube videos</td>
<td>www you tube com</td>
<td>www www youtube com</td>
</tr>
<tr>
<td>www youtube</td>
<td>www youtube com</td>
<td>www youtube co</td>
</tr>
<tr>
<td>yotube</td>
<td>www you tube</td>
<td>www u tube com</td>
</tr>
<tr>
<td>ww youtube com</td>
<td>www utube</td>
<td>www u tube</td>
</tr>
<tr>
<td>utube videos</td>
<td>utube com</td>
<td>utube</td>
</tr>
<tr>
<td>u tube com</td>
<td>utub</td>
<td>u tube videos</td>
</tr>
<tr>
<td>u tube</td>
<td>my tube</td>
<td>toutube</td>
</tr>
<tr>
<td>outube</td>
<td>our tube</td>
<td>toutube</td>
</tr>
</tbody>
</table>
Matching at Different Levels

- **Term**: Match between terms in query & document
  - **NY** → **NY**
  - **youtube** → **youtube**

- **Phrase**: Match between phrases in query & document
  - **hot dog** → **hot dog**

- **Word Sense**: Match between word senses in query & document
  - **youtube** → **youtube**
  - **NY** → **New York**

- **Topic**: Match between topics of query & document
  - **Microsoft Office** → **Microsoft PowerPoint, Word, Excel**

- **Structure**: Match between structures of query & document title
  - **how far is sun from earth** → **distance between sun and earth**
Query Understanding

1. Structure Identification
2. Topic Identification
3. Similar Query Finding
4. Phrase Identification
5. Spelling Error Correction

michael jordan: main phrase
michael jordan berkely: machine learning
michael l. jordan
michael jordan
[michael jordan] berkeley
michael jordan berkeley
Homepage of Michael Jordan

Michael Jordan is Professor in the Department of Electrical Engineering

Michael Jordan: main phrase in Title

Michael Jordan is Professor in the Department of Electrical Engineering: machine learning

[Michael Jordan], [Professor] [Electrical Engineering]: keyphrase

[Michael Jordan] is [Professor] in the [Department] of [Electrical Engineering]
Online Matching

Matching can be conducted at different levels

Query Representation → Semantic Matching → Document Representation

Ranking Result

Michael Jordan's Home Page
Models of visuomotor and other learning (Univ. of California, Berkeley, USA)
www.cs.berkeley.edu/~jordan · Cached page · Mark as spam

Michael Jordan | EECS at UC Berkeley
Michael Jordan Professor Research Areas Artificial Intelligence (AI) Biomedical Engineering
Computational Biology (BIO) Control, Intelligent Systems, and Robotics (CR)
www.eecs.berkeley.edu/Faculty/Homepages/jordan.html · Cached page · Mark as spam

Publications:
www.cs.berkeley.edu/~jordan/publications.html · Cached page · Mark as spam
Relation between Matching and Ranking

• In traditional IR:
  – Ranking = matching
    \[ f(q,d) = f_{BM25}(q,d) \text{ or } f(q,d) = P_{LMIR}(d \mid q) \]

• Web search:
  – Ranking and matching become separated
  – Learning to rank becomes state-of-the-art
    \[ f(q,d) = f_{BM25}(q,d) + g_{PageRank}(d) + \cdots \]
  – Matching = feature learning for ranking
Matching Functions as Features in Learning to Rank

- Term level matching: $f_{BM25}(q, d) \quad f_{n-BM25}(q, d)$
- Phrase level matching: $f_P(q, d)$
- Sense level matching: $f_S(q, d)$
- Topic level matching: $f_T(q, d)$
- Structure level matching: $f_C(q, d)$
- Term level matching (spelling, stemming): $q' \rightarrow q$
Previous Work

• Studied in long history of IR
• Query expansion, pseudo relevance feedback
• Latent Semantic Indexing, Probabilistic Latent Semantic Indexing
• ... ...
## Previous Work v.s. Recent Work

<table>
<thead>
<tr>
<th></th>
<th>Previous</th>
<th>Recent</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scale</td>
<td>Small</td>
<td>Large</td>
</tr>
<tr>
<td>Methodologies</td>
<td>Unsupervised learning</td>
<td>Both supervised learning and unsupervised learning</td>
</tr>
<tr>
<td>Data</td>
<td>No use of log data</td>
<td>Use of log data</td>
</tr>
</tbody>
</table>
Regularized Latent Semantic Indexing

Joint Work with Quan Wang, Jun Xu, and Nick Craswell
Regularized Latent Semantic Indexing
(Wang et al. SIGIR 2011)

• Motivation
  – Matching between query and document at topic level
  – Scale up to large datasets (vs. existing methods)

• Approach
  – Matrix Factorization
  – Regularization on topics and documents (vs. Sparse Coding)
  – Learning problem can be easily parallelized

• Results
  – $l_1$ on topics leads to sparse topics and $l_2$ on documents leads to accurate matching
  – Comparable with existing methods in topic discovery and search relevance
  – But can easily scale up to large data sets
Query and Document Matching in Topic Space

Document Space

Topic Space
Regularized Latent Semantic Indexing

document representation of doc $n$

\[
\min_{U, \{v_n\}} \sum_{n=1}^{N} \|d_n - Uv_n\|_2^2 + \lambda_1 \sum_{k=1}^{K} \|u_k\|_1 + \lambda_2 \sum_{n=1}^{N} \|v_n\|_2^2
\]
Optimization Strategy

\[ \min_{\mathbf{U}, \{v_n\}} \sum_{n=1}^{N} \left\| \mathbf{d}_n - \mathbf{U}v_n \right\|_2^2 + \lambda_1 \sum_{k=1}^{K} \left\| \mathbf{u}_k \right\|_1 + \lambda_2 \sum_{n=1}^{N} \left\| \mathbf{v}_n \right\|_2^2 \]

Coordinate Decent

\[ \min_{\{\bar{u}_m\}} \sum_{m=1}^{M} \left\| \bar{d}_m - \mathbf{V}^T \bar{u}_m \right\|_2^2 + \lambda_1 \sum_{m=1}^{M} \left\| \bar{u}_m \right\|_1 \]

for \( m = 1, \cdots, M \)

\[ u_{mk} = \begin{cases} 
\frac{(r_{mk} - \sum_{l \neq k} s_{kl} u_{ml}) - \frac{1}{2} \lambda_1}{s_{kk}}, & \text{if } u_{mk} > 0 \\
\frac{s_{kk} (r_{mk} - \sum_{l \neq k} s_{kl} u_{ml}) + \frac{1}{2} \lambda_1}{s_{kk}}, & \text{if } u_{mk} < 0 
\end{cases} \]

Update \( \mathbf{U} \)

Update \( \mathbf{V} \)

Analytic Solution

\[ v_n^* = \left( \mathbf{U}^T \mathbf{U} + \lambda_2 \mathbf{I} \right)^{-1} \mathbf{U}^T \mathbf{d}_n \]
RLSI Algorithm

• Single machine multi core version
• Multiple machine version (MapReduce and MPI)

Algorithm 2 Update U

Require: $D \in \mathbb{R}^{M \times N}, V \in \mathbb{R}^{K \times N}$
1: $S \leftarrowVV^T$
2: $R \leftarrow DV^T$
3: for $m = 1 : M$ do
   4: $\tilde{u}_m \leftarrow 0$
   5: repeat
   6: for $k = 1 : K$ do
      7: $w_{mk} \leftarrow r_{mk} - \sum_{l \neq k} s_{kl}u_{ml}$
      8: $u_{mk} \leftarrow \left(\frac{|w_{mk}| - \frac{1}{2} AN}{s_{kk}}\right) \text{sign}(w_{mk})$
   9: end for
10: until convergence
11: end for
12: return $U$

Algorithm 3 Update V

Require: $D \in \mathbb{R}^{M \times N}, U \in \mathbb{R}^{M \times K}$
1: $\Sigma \leftarrow (U^T U + \theta I)^{-1}$
2: $\Phi \leftarrow U^T D$
3: for $n = 1 : N$ do
   4: $v_n \leftarrow \Sigma \phi_n$, where $\phi_n$ is the $n^{th}$ column
   5: end for
6: return $V$
## Scalability Comparison

<table>
<thead>
<tr>
<th>algorithm</th>
<th>max dataset applied (#docs; #words)</th>
<th># topics</th>
<th># processors used</th>
</tr>
</thead>
<tbody>
<tr>
<td>PLDA and PLDA+ (by Google)</td>
<td>Wiki-200T(2,112,618; 200,000)</td>
<td>1000</td>
<td>2,048</td>
</tr>
<tr>
<td>AD-LDA</td>
<td>NY Times (300,000; 102,660)</td>
<td>200</td>
<td>16</td>
</tr>
<tr>
<td></td>
<td>PubMed (8,200,000; 141,043)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RLSI</td>
<td>B01 (1,562,807; 7,014,881)</td>
<td>500 ~ 1000</td>
<td>single machine, 24 cores</td>
</tr>
<tr>
<td></td>
<td>Bing News (940,702; 500,033)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Wiki-All (3,239,884; 6,043,069)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>MSWeb Data (2,635,158; 2,371,146)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Experimental Results on Topic Discovery

Topics discovered by RLSI are equally readable compared with LDA, PLSI, LSI

<table>
<thead>
<tr>
<th>Topic 1</th>
<th>Topic 2</th>
<th>Topic 3</th>
<th>Topic 4</th>
<th>Topic 5</th>
<th>Topic 6</th>
<th>Topic 7</th>
<th>Topic 8</th>
<th>Topic 9</th>
<th>Topic 10</th>
</tr>
</thead>
<tbody>
<tr>
<td>oil</td>
<td>south</td>
<td>infect</td>
<td>test</td>
<td>noriega</td>
<td>percent</td>
<td>plane</td>
<td>israeli</td>
<td>nuclear</td>
<td>bush</td>
</tr>
<tr>
<td>cent</td>
<td>african</td>
<td>test</td>
<td>educate</td>
<td>panama</td>
<td>billion</td>
<td>crash</td>
<td>palestinian</td>
<td>soviet</td>
<td>dukakis</td>
</tr>
<tr>
<td>barrel</td>
<td>angola</td>
<td>patient</td>
<td>college</td>
<td>panamanian</td>
<td>rate</td>
<td>flight</td>
<td>israel</td>
<td>campaign</td>
<td>quayle</td>
</tr>
<tr>
<td>price</td>
<td>apartheid</td>
<td></td>
<td></td>
<td>delval</td>
<td>0</td>
<td>air</td>
<td>arab</td>
<td>weapon</td>
<td>bentsen</td>
</tr>
</tbody>
</table>

Table 8: Topics discovered by RLSI, LDA, PLSI, and LSI from AP dataset.
Experimental Results on Web Search

Reducing vocabulary hurts ranking accuracy.

RLSI can help improve search relevance.
Online Regularized Latent Semantic Indexing (Wang et al., under review)

- Text data is in stream
- Online learning of topic model
- Can scale up to large datasets with small storage
- Can capture dynamics of text stream

\[ d_1 \sim d_{t-1}, \quad \nu_1 \sim \nu_{t-1}, \quad U_{t-1} \rightarrow d_t, \quad \nu_t, \quad U_t \]
Group Regularized Latent Semantic Indexing (Wang et al. SIGIR 2012)

- Assuming that documents have been classified into classes
- Assuming that there exist class specific topics and shared topics
- Can scale up to even large data sets

\[ D = \]

\[ D(1) \approx U(0) U(1) \times V(1) \]

\[ D(2) \approx U(0) U(2) \times V(2) \]
Matching in Latent Space

Joint Work with Wei Wu, Jun Xu, Zhengdong Lv
Matching in Latent Space
(Wu et al., WSDM 2013, Wu et al., under review)

• Motivation
  – Matching between query and document in latent space

• Assumption
  – Queries have similarity
  – Document have similarity
  – Click-through data represent “similarity” relations between queries and documents

• Approach
  – Projection to latent space
  – Regularization or constraints

• Results
  – Significantly enhance accuracy of query document matching
Matching in Latent Space

Query Space

Document Space

Latent Space

$q_1$, $q_2$, $d_1$, $d_2$, $q_m$, $d_n$
Example: Projecting Keywords and Images into Latent Space
Partial Least Square (PLS)

- **Setting**
  - Two spaces: $\mathcal{X} \subset \mathbb{R}^m$ and $\mathcal{Y} \subset \mathbb{R}^n$.

- **Input**
  - Training data: $\{(x_i, y_i, r_i)\}_{1 \leq i \leq N}$, $r_i \in \{+1, -1\}$ (or $r_i \in \mathbb{R}$)

- **Output**
  - Similarity function $f(x, y)$

- **Assumption**
  - Two linear (and orthonormal) transformations $L_X$ and $L_Y$
  - Dot product as similarity function $\langle L_X^T x, L_Y^T y \rangle = x^T L_X L_Y^T y$

- **Optimization**
  $$\arg \max_{L_X, L_Y} \sum_{r_i = +1} x_i^T L_X L_Y^T y_i - \sum_{r_i = -1} x_i^T L_X L_Y^T y_i$$
  subject to $L_X^T L_X = I_{k \times k}, L_Y^T L_Y = I_{k \times k}$
Solution of Partial Least Square

• Non-convex optimization
• Can prove that global optimal solution exists
• Global optimal can be found by solving SVD (Singular Value Decomposition)
• SVD of Matrix $M_S - M_D = U \Sigma V^T$
Regularized Mapping to Latent Space (RMLs)

- **Setting**
  - Two spaces: \( \mathcal{X} \subset \mathbb{R}^m \) and \( \mathcal{Y} \subset \mathbb{R}^n \).

- **Input**
  - Training data: \( \{(x_i, y_i, r_i)\}_{1 \leq i \leq N} \), \( r_i \in \{+1, -1\} \) (or \( r_i \in \mathbb{R} \))

- **Output**
  - Similarity function \( f(x, y) \)

- **Assumption**
  - L1 and L2 regularization on \( L_x \) and \( L_y \) (sparse transformations)
  - Dot product as similarity function \( \langle L_x^T x, L_y^T y \rangle = x^T L_x L_y^T y \)

- **Optimization**
  \[
  \arg\max_{L_x, L_y} \sum_{r_i=+1} x_i^T L_x L_y^T y_i - \sum_{r_i=-1} x_i^T L_x L_y^T y_i
  \]

  subject to \( |l_x| \leq \vartheta x, |l_y| \leq \vartheta y, \| l_x \| \leq \lambda x, \| l_y \| \leq \lambda y, \)
Solution of Regularized Mapping to Latent Space

- Coordinate Descent
- Repeat
  - Fix $L_x$, update $L_y$
  - Fix $L_y$, update $L_x$
- Update can be parallelized by rows
## Comparison

<table>
<thead>
<tr>
<th></th>
<th>PLS</th>
<th>RMLSS</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Assumption</strong></td>
<td>Orthogonal</td>
<td>L1 and L2 Regularization</td>
</tr>
<tr>
<td><strong>Optimization Method</strong></td>
<td>Singular Value Decomposition</td>
<td>Coordinate Descent</td>
</tr>
<tr>
<td><strong>Optimality</strong></td>
<td>Global optimum</td>
<td>Local optimum</td>
</tr>
<tr>
<td><strong>Efficiency</strong></td>
<td>Low</td>
<td>High</td>
</tr>
<tr>
<td><strong>Scalability</strong></td>
<td>Low</td>
<td>High</td>
</tr>
</tbody>
</table>
Experimental Results

<table>
<thead>
<tr>
<th>Enterprise Search</th>
<th>NDCG@1</th>
<th>NDCG@3</th>
<th>NDCG@5</th>
</tr>
</thead>
<tbody>
<tr>
<td>MPLS&lt;sub&gt;Com&lt;/sub&gt;</td>
<td>0.715</td>
<td>0.733</td>
<td>0.747</td>
</tr>
<tr>
<td>MPLS&lt;sub&gt;Conca&lt;/sub&gt;</td>
<td>0.700</td>
<td>0.728</td>
<td>0.742</td>
</tr>
<tr>
<td>MPLS&lt;sub&gt;Word&lt;/sub&gt;</td>
<td>0.688</td>
<td>0.718</td>
<td>0.739</td>
</tr>
<tr>
<td>MPLS&lt;sub&gt;Bipar&lt;/sub&gt;</td>
<td>0.659</td>
<td>0.684</td>
<td>0.705</td>
</tr>
<tr>
<td>BM25</td>
<td>0.653</td>
<td>0.657</td>
<td>0.663</td>
</tr>
<tr>
<td>RW</td>
<td>0.654</td>
<td>0.683</td>
<td>0.700</td>
</tr>
<tr>
<td>RW+BM25</td>
<td>0.664</td>
<td>0.688</td>
<td>0.705</td>
</tr>
<tr>
<td>LSI</td>
<td>0.656</td>
<td>0.676</td>
<td>0.695</td>
</tr>
<tr>
<td>LSI+BM25</td>
<td>0.692</td>
<td>0.701</td>
<td>0.712</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Web Search</th>
<th>NDCG@1</th>
<th>NDCG@3</th>
<th>NDCG@5</th>
</tr>
</thead>
<tbody>
<tr>
<td>MPLS&lt;sub&gt;Com&lt;/sub&gt;</td>
<td>0.681</td>
<td>0.731</td>
<td>0.739</td>
</tr>
<tr>
<td>MPLS&lt;sub&gt;Conca&lt;/sub&gt;</td>
<td>0.676</td>
<td>0.728</td>
<td>0.736</td>
</tr>
<tr>
<td>MPLS&lt;sub&gt;Word&lt;/sub&gt;</td>
<td>0.674</td>
<td>0.726</td>
<td>0.732</td>
</tr>
<tr>
<td>MPLS&lt;sub&gt;Bipar&lt;/sub&gt;</td>
<td>0.612</td>
<td>0.680</td>
<td>0.693</td>
</tr>
<tr>
<td>BM25</td>
<td>0.637</td>
<td>0.690</td>
<td>0.690</td>
</tr>
<tr>
<td>RW</td>
<td>0.655</td>
<td>0.704</td>
<td>0.704</td>
</tr>
<tr>
<td>RW+BM25</td>
<td>0.671</td>
<td>0.718</td>
<td>0.716</td>
</tr>
<tr>
<td>LSI</td>
<td>0.588</td>
<td>0.665</td>
<td>0.676</td>
</tr>
<tr>
<td>LSI+BM25</td>
<td>0.649</td>
<td>0.705</td>
<td>0.706</td>
</tr>
</tbody>
</table>

- RMLS and PLS work better than BM25, Random Walk, Latent Semantic Indexing
- RMLS works equally well as PLS, with higher learning efficiency and scalability
Projecting to Latent Space with Multi-View Learning (Wu et al. WSDM 2013)

- Multi-view PLS
- Global optimum is guaranteed

\[ \begin{align*}
q_1 & \quad q_m & \quad q_2 \\
\text{Query Space} & & \text{Latent Space} \\
\end{align*} \]

\[ \begin{align*}
d_1 & \quad d_n \\
\text{Document Space} & & \text{Latent Space} \\
\end{align*} \]
Learning Matching Model with Kernel Method
(Wu et al., JMLR 2011)

• Learning robust matching model (IR model)
• Using k-nearest neighbors to smooth
• Approach: Kernel method

Hilbert space \( k_Q(q, q') \)
Hilbert space \( k_{IR}(q, d) \)
Hilbert space \( k_{IR}(q', d') \)
Hilbert space \( k_D(d, d') \)

Query space
Document space

Query-document pair space

Similarity Functions

\( \hat{k} ((q, d), (q', d')) \)
String Re-writing Kernel

Joint work with Fan Bu and Xiaoyan Zhu
String Re-writing Kernel
(Bu et al. ACL 2012, best student paper award)

• Motivation
  – Matching between two strings at structure level
  – High flexibility

• Approach
  – A new class of kernel functions (SRK)
  – A subclass kb-SRK, which can be efficiently computed

• Results
  – Outperforms state of the art methods in paraphrasing and entailment on benchmark datasets
Learning with String Re-wring Kernel

((distance between sun and earth), (how far sun earth), +1)
((distance between beijing and shanghai), (how far is beijing from shanghai), +1)
……
((distance between moon and earth), (how far sun earth), -1)

Learning System

Prediction System

+1/-1?

Model

((distance between moon and earth), (how far is earth from moon))
Problem Formulation

• Training data

\[((s_1, t_1), y_1) \cdots ((s_n, t_n), y_n)\]

• Model

\[y = \text{sign} \left( \sum_{i=1}^{n} \alpha_i y_i K((s_i, t_i), (s, t)) \right) \quad \alpha_i \geq 0\]

• String Re-writing Kernel

\[K((s_i, t_i), (s, t))\]
Re-Writing Rule

- Rule is applied to re-writing of substrings

Shakespeare wrote Hamlet in 16 century

Hamlet was written by Shakespeare

Cao Xueqin wrote Dream of the Red Chamber

Dream of the Red Chamber was written by Cao Xueqin

Cao Xueqin wrote Dream of the Red Chamber

Hamlet was written by Shakespeare
Multiple re-writing rules may be applied

- Shakespeare wrote Hamlet in 16 century
- Hamlet was written by Shakespeare

Re-writing Rule

* wrote * in 16 century
* was written by *
Formulation of String Re-writing Kernel (SRK)

\[ K((s_1, t_1), (s_2, t_2)) = \langle \Phi(s_1, t_1), \Phi(s_2, t_2) \rangle \]

\[ \Phi(s, t) = (\phi_r(s, t))_{r \in R} \]

\[ \phi_r(s, t) = n\lambda^i \quad \lambda \in (0, 1] \]
Similarity between Two Re-writings

- Similarity between two re-writings with respect to re-writing rules

Shakespeare wrote *Hamlet* in 16 century

Hamlet was written by Shakespeare

Cao Xueqin wrote *Dream of the Red Chamber*

Dream of the Red Chamber was written by Cao Xueqin

$$\langle r_1 = 0, \ldots, r_{100} = 1, \ldots, r_{1001} = 1, \ldots \rangle$$   $$\langle r_1 = 0, \ldots, r_{100} = 1, \ldots, r_{1001} = 0, \ldots \rangle$$
String Re-writing Kernel

• Advantage: Matching between informally written sentences such as long queries in search can be effectively performed

• Challenge
  • Number of re-writing rules is infinite
  • Number of applicable rules increase exponentially when length of sentence increases

• Our Approach
  • Sub-class: kb-SRK
Definition of kb-SRK

- Special class of SRK
- Re-writing rules in kb-SRK
  - String patterns in rule are of length k
  - Wildcard ? only substitutes a single character
  - Alignment between string patterns is bijective

? wrote ?
? described ?
Formulation of kb-SRK

\[ K_k (\langle s_1, t_1 \rangle, \langle s_2, t_2 \rangle) = \sum_{\alpha_{s_1} \in k\text{-gram}(s_1)} \sum_{\alpha_{t_1} \in k\text{-gram}(t_1)} \sum_{\alpha_{s_2} \in k\text{-gram}(s_2)} \sum_{\alpha_{t_2} \in k\text{-gram}(t_2)} K_k (\langle \alpha_{s_1}, \alpha_{t_1} \rangle, \langle \alpha_{s_2}, \alpha_{t_2} \rangle) \]

\[ \overline{K}_k = \sum_{r \in R} \overline{\phi}_r (\alpha_{s_1}, \alpha_{t_1}) \overline{\phi}_r (\alpha_{s_2}, \alpha_{t_2}) \]
Efficiently Calculating Number of Applicable Rules on Combined k-gram Pairs

two k-gram pairs

\[ \alpha_{s_1} = \text{abcccb} \quad \alpha_{s_2} = \text{abcccd} \]
\[ \alpha_{t_1} = \text{cbccbc} \quad \alpha_{t_2} = \text{cbbccd} \]

combined k-gram pairs

\[ \alpha_s = (a, a)(b, b)(b, c)(c, c)(c, c)(b, d)(b, d) \]
\[ \alpha_t = (c, c)(b, b)(c, c)(b, c)(b, d)(c, c)(b, d) \]

applicable rule

\[ \text{ab?cc??} \]
\[ \text{cbc??c?} \]

applicable rule

\[ (a, a)(b, b) \quad ?(c, c)(c, c) \quad ? \]
\[ (c, c)(b, b)(c, c) \quad ?(c, c) \quad ? \]

Time complexity, empirically

\[ O(k \cdot n^2) \]
\[ n = \max(|s_1|, |t_1|, |s_2|, |t_2|) \]

identical doubles can be either or not substituted
non-identical doubles must be substituted
Experiment: Paraphrase Identification

- Comparison with state-of-the-arts methods.

<table>
<thead>
<tr>
<th>Method</th>
<th>Acc.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Zhang and Patrick (2005)</td>
<td>71.9</td>
</tr>
<tr>
<td>Linteau and Rus (2011)</td>
<td>73.6</td>
</tr>
<tr>
<td>Heilman and Smith (2010)</td>
<td>73.2</td>
</tr>
<tr>
<td>Qiu et al. (2006)</td>
<td>72.0</td>
</tr>
<tr>
<td>Wan et al. (2006)</td>
<td>75.6</td>
</tr>
<tr>
<td>Das and Smith (2009)</td>
<td>73.9</td>
</tr>
<tr>
<td>Das and Smith (2009)(PoE)</td>
<td>76.1</td>
</tr>
<tr>
<td>Our baseline (PR)</td>
<td>73.6</td>
</tr>
<tr>
<td>Our method (ps-SRK)</td>
<td>75.6</td>
</tr>
<tr>
<td>Our method (pw-SRK)</td>
<td>75.0</td>
</tr>
<tr>
<td>Our method (kb-SRK)</td>
<td>76.3</td>
</tr>
</tbody>
</table>
Experiment: Recognizing Textual Entailment

- Comparison with state-of-the-arts methods.

<table>
<thead>
<tr>
<th>Method</th>
<th>Acc.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Harmeling (2007)</td>
<td>59.5</td>
</tr>
<tr>
<td>de Marneffe et al. (2006)</td>
<td>60.5</td>
</tr>
<tr>
<td>M&amp;M, (2007) (NL)</td>
<td>59.4</td>
</tr>
<tr>
<td>M&amp;M, (2007) (Hybrid)</td>
<td>64.3</td>
</tr>
<tr>
<td>Zanzotto et al. (2007)</td>
<td>65.75</td>
</tr>
<tr>
<td>Heilman and Smith (2010)</td>
<td>62.8</td>
</tr>
<tr>
<td>Our baseline (PR)</td>
<td>62.0</td>
</tr>
<tr>
<td>Our method (ps-SRK)</td>
<td>64.6</td>
</tr>
<tr>
<td>Our method (pw-SRK)</td>
<td>63.8</td>
</tr>
<tr>
<td>Our method (kb-SRK)</td>
<td>65.1</td>
</tr>
</tbody>
</table>
Summary
Open Questions in Learning to Match

- What are the relations between existing methods?
- Which methods are more suitable for which tasks?
- What are the theoretical properties of existing methods?
- Is there a single framework that can formalize all existing methods?
- Is it possible to combine existing methods?
- How to scale up existing methods to large datasets?
- How to extend existing methods to handle more complicated tasks?
- How to clean up the data if it is from user feedbacks?
Summary

• Matching - Fundamental Problem in Many Applications
• Learning to Match
• Learning to Match is Key Issue in Search
• Approaches
  – Regularized Latent Semantic Indexing
  – Matching in Latent Space
  – String Re-writing Kernel
• More Research on Learning to Match Is Necessary
Publications

- Wei Wu, Zhengdong Lv, Hang Li, Regularized Mapping to Latent Structures and Its Application to Web Search, submitted to JMLR.
- Ziqi Wang, Gu Xu, Hang Li, Ming Zhang, A Statistical Learning Approach to String Transformation, Submitted to TKDE.
- Wei Wu, Hang Li, Learning Query and Document Similarities from Click-through Bipartite Graph with Metadata, WSDM 2013, to appear.
Publications

• Gu Xu, Shuanghong Yang, Hang Li, Named Entity Mining from Click-Through Log Using Weakly Supervised Latent Dirichlet Allocation. In Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'09), 1365-1374.
Thank You!

Contact: hangli.hl@huawei.com