The Natural Language Challenge in Web Search

Hang Li
Noah’s Ark Lab
Huawei Technologies
Biggest Challenge in Web Search is Natural Language Challenge
Natural Language Challenge = Deal with Query Document Mismatch
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 from the sun to the 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 from the sun
- how far earth is from the sun
- how far from earth is the sun
- how far from earth to sun
- how far from the earth to the sun
- distance between sun and earth
Same Search Intent, Different Query Representations
Example = “Youtube”

- yutube
- ytube
- youtubo
- youtube om
- youtube
- youtub com
- youtub
- you tube
- you tube videos
- www you tube
- yotube
- ww youtube com
- utube videos
- u tube com
- u tube
- outube

- yuotube
- youtubr
- youtuber
- youtube music videos
- youtube com
- you tube music videos
- you tube com yourtube
- you tub
- www you tube com
- www you tube
- www you tube
- www utube
- www utube com
- utube com
- utub
- my tube
- our tube

- yuo tube
- yu tube
- youtubecom
- youtube videos
- youtube co
- your tube
- you tube video clips
- wwwww youtube com
- www youtube co
- www utube com
- utube
- www u tube
- utube
- u tube videos
- toutube
- toutube
## Examples of Query Document Mismatch

<table>
<thead>
<tr>
<th>Query</th>
<th>Document</th>
<th>Term Matching</th>
<th>Semantic Matching</th>
</tr>
</thead>
<tbody>
<tr>
<td>seattle best hotel</td>
<td>seattle best hotels</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>pool schedule</td>
<td>swimmingpool schedule</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>natural logarithm transformation</td>
<td>logarithm transformation</td>
<td>partial</td>
<td>yes</td>
</tr>
<tr>
<td>china kong</td>
<td>china hong kong</td>
<td>partial</td>
<td>no</td>
</tr>
<tr>
<td>why are windows so expensive</td>
<td>why are macs so expensive</td>
<td>partial</td>
<td>no</td>
</tr>
</tbody>
</table>
Talk Outline

1. Machine Learning for Query Document Matching
2. Matching in Latent Space
3. QRU-1 Dataset
1. Machine Learning for Query Document Matching in Web Search
Matching at Different Levels

- **Match between structures of query & document title**
  - how far is sun from earth → ... distance between sun and earth

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

- **Match between word senses in query & document**
  - uTube → youtube  NY → New York

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

- **Match between terms in query & document**
  - NY → NY  youtube → youtube
Query Understanding

- Structure Identification
- Topic Identification
- Similar Query Finding
- Phrase Identification
- Spelling Error Correction

michael jordan: main phrase

michael jordan berkely: machine learning

michael l. jordan
michael jordan

[michael jordan] berkeley

michael jordan berkele
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
Learning for Matching between Query and Document

- Learning matching function
  \[ f_M(q,d) \text{ or } p_M(r \mid q,d) \]

- Using training data \((q_1,d_1,r_1), \ldots, (q_N,d_N,r_N)\)
  - \(q_1, q_2, \ldots, q_N\) and \(d_1, d_2, \ldots, d_N\) can be id’s or feature vectors
  - \(r_1, r_2, \ldots, r_N\) can be binary or numerical values

- Using relations in data and/or prior knowledge
Matching Problem: Instance Matching

Graph View

q1

q2

q3

qm

1 3 5

d1 d2 d3

qn
Matching Problem: Instance Matching

Matrix View

<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></td>
</tr>
<tr>
<td>q2</td>
<td></td>
<td>4</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>q3</td>
<td></td>
<td></td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>qm</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Matching Problem: Content Matching

Query space

Document space

Space View
## Matching Problem: Content Matching

### Matrix View

<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></td>
<td>1</td>
<td></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>1</td>
<td>5</td>
<td></td>
</tr>
</tbody>
</table>
Challenges in Machine Learning for Matching

• How to leverage relations in data and prior knowledge

• Scale is very large
Relation between Matching and Ranking

• In traditional IR:
  – Ranking = matching

\[ f(q, d) = f_{BM25}(q, d) \quad \text{or} \quad f(q, d) = P_{LMIR}(d \mid q) \]

• 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 v.s. Ranking

<table>
<thead>
<tr>
<th></th>
<th>Matching</th>
<th>Ranking</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Prediction</strong></td>
<td>Matching score between query and document</td>
<td>List of documents</td>
</tr>
<tr>
<td><strong>Model</strong></td>
<td>$f(q, d)$</td>
<td>$f(q,d1), f(q,d2), \ldots, f(q,dn)$</td>
</tr>
<tr>
<td><strong>Loss Function</strong></td>
<td>Query document pair</td>
<td>List of documents with respect to query</td>
</tr>
<tr>
<td><strong>Challenge</strong></td>
<td>Mismatch</td>
<td>Correct ranking on top</td>
</tr>
</tbody>
</table>
Previous Work

• Studied in long history of IR
• Query expansion, pseudo relevance feedback
• Latent Semantic Indexing, Probabilistic Latent Semantic Indexing
• … …
New Trends in Recent Work

• Employing more machine learning (supervised and unsupervised)
• Large scale
• Use of log data

• This tutorial focuses on recent work!
# 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>
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): $q' \rightarrow q$
Linear Combinations of Matching Functions

• Query Reformulation

\[ f(q, d) = f_{BM25}(q, d) + \sum_{i} k_Q(q, q_i)k_D(d, d_i) f(q_i, d_i) \]

• Topic Model

\[ f(q, d) = f_{LMIR}(q, d) + \sum_{k} u(q, k)v(k, d) \]
Approaches to Learning for Matching Between Query and Document

- Matching by Query Reformulation
- Matching with Dependency Model
- Matching with Topic Model
- Matching with Translation Model
- Matching in Latent Space
2. Matching in Latent Space

Joint Work with Wei Wu, Jun Xu, Zhengdong Lv
Matching in Latent Space

• Queries have similarity
• Document have similarity
• Queries and documents are heterogeneous data
• Click-through data represent “similarity” relations between queries and documents
Matching in Latent Space (2)

• Question:
  – Preserve query similarity
  – Preserve document similarity
  – Calculate query document similarity from click-through data

• Solutions
  – Partial Least Square (Wu et al., ’11)
  – Regularized Mapping to Latent Space (Wu et al., 12’)

Mapping into Latent Space

Query Space  New Space  Document Space

$q_1, q_2, q_m$  $d_1, d_2, d_n$  $L_q, L_d$
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: $\{(x_i, y_i, r_i)\}_{1 \leq i \leq N}$, $r_i \in \{+1, -1\}$

- **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: $\{(x_i, y_i, r_i)\}_{1 \leq i \leq N}$, $r_i \in \{+1, -1\}$

- **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 $|lx| \leq \vartheta x$, $|ly| \leq \vartheta y$, $\|lx\| \leq \lambda x$, $\|ly\| \leq \lambda y$, $RML$
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>RMLs</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>Web Search</th>
</tr>
</thead>
<tbody>
<tr>
<td>NDCG@1</td>
<td>NDCG@3</td>
</tr>
<tr>
<td>MPLS&lt;sub&gt;Com&lt;/sub&gt;</td>
<td>0.715</td>
</tr>
<tr>
<td>MPLS&lt;sub&gt;Conca&lt;/sub&gt;</td>
<td>0.700</td>
</tr>
<tr>
<td>MPLS&lt;sub&gt;Word&lt;/sub&gt;</td>
<td>0.688</td>
</tr>
<tr>
<td>MPLS&lt;sub&gt;Bipar&lt;/sub&gt;</td>
<td>0.659</td>
</tr>
<tr>
<td>BM25</td>
<td>0.653</td>
</tr>
<tr>
<td>RW</td>
<td>0.654</td>
</tr>
<tr>
<td>RW+BM25</td>
<td>0.664</td>
</tr>
<tr>
<td>LSI</td>
<td>0.656</td>
</tr>
<tr>
<td>LSI+BM25</td>
<td>0.692</td>
</tr>
</tbody>
</table>

- Latent Space Model (Multi-view PLS) works better than BM25, Random Walk, Latent Semantic Indexing
- RMLS works equally well as PLS, with higher learning efficiency and scalability
3. QRU-1 Dataset

Joint Work with Michael Bendersky, Gu Xu, Bruce Croft

Downloadable at MSR Web Site

bit.ly/qru1dataset
Motivation for QRU-1

• Benchmark dataset for research on query transformation, etc
• Queries are as real as possible
• Queries are related to existing benchmark datasets (e.g., TREC query sets) for better connection with existing work
Content of Dataset

• Seed: 100 queries from TREC Web Track (2009 and 2010)
• Each query is assigned similar queries (on average 20 queries)
• Similar queries represent the same or similar search intents as original queries
• Similar queries may contain typos, stemming, synonyms
• In total, 2036 similar queries
Examples of Similar Queries

95: earn money at home
- earn money from home
- earn money at home
- how to earn money at home
- earn money on the internet
- ways to earn money at home
- how to earn money from home
- earn extra money at home
- earning money from home
- earn extra cash at home
- earning money at home
- earn at home
- earn money working from home
- earn money from home free
- how to earn money on the internet
- earn cash at home
- earn currency at home
- earn money at home
- earn money at hoem
Process of Data Creation

• Obtained 100 TREC queries
• Trained a query generation model using the method by (Wang et al. 2011) and search log data at Bing (2010/07-2010/12)
• Generated similar queries from TREC queries with the model
• Manually removed mistakenly generated queries (23% of generated queries were removed)
• Observed about 70% of the generated queries actually exist in real Bing log data
• Got approval for release from MS legal team
Learning of Generation Model

Training Data

\((q^1_m, q^1_c)\)

\((q^2_m, q^2_c)\)

\((q^3_m, q^3_c)\)

\(\ldots\)

Rule Extraction

\(\alpha_1 \rightarrow \beta_1\)

\(\alpha_2 \rightarrow \beta_2\)

\(\alpha_3 \rightarrow \beta_3\)

\(\ldots\)

Model Learning

\(P(q_c, R(q_m, q_c) \mid q_m)\)

log linear model

Model

\(\alpha_1 \rightarrow \beta_1, \lambda_1\)

\(\alpha_2 \rightarrow \beta_2, \lambda_2\)

\(\alpha_3 \rightarrow \beta_3, \lambda_3\)

\(\ldots\)

weight
Rule Extraction

• Edit-distance based alignment:

  \[ \begin{array}{ccc}
  \text{Original:} & ^{\_} & a & b & c & \$ \\
  \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\
  \text{Similar:} & ^{\_} & a & e & c & \$ \\
  \end{array} \]

• Basic substitution rules:

  \[ b \rightarrow e \]

• Contextual substitution rules

  \[ ^{\_}ab \rightarrow ^{\_}ae, \ ab \rightarrow ae, \ bc \rightarrow ec, \ldots \]
Log Linear Model

• Model

\[ P(q_c, R(q_m, q_c) | q_m) = \frac{\exp(\sum_{r \in R(q_m, q_c)} \lambda_r)}{\sum_{(q'_c, R(q_m, q'_c)) \in Z(q_m)} \exp(\sum_{o \in R(q_m, q'_c)} \lambda_o)} \]

Set of rules rewrite \( q_m \) to \( q_c \)

Weight of rule

All pairs of word \( q'_c \) and rule set \( R(q_m, q'_c) \)

• Candidate Generation

\[ \text{rank}(q_c | q_m) = \max_{R(q_m, q_c)} (\sum_{r \in R(q_m, q_c)} \lambda_r) \]
Guidelines for Manual Cleaning

• Keep generated queries, if
  – they represent the same intents as the original queries, and
  – they are likely to be input by users, including typos

• Otherwise discard the queries
  – E.g. “pictures of the obama family”
  – E.g. “obama family plant”
  – E.g. “michelle obama family tree”
One Possible Way of Using The Data

• Assuming similar queries are submitted by users
• Conducting retrieval and ranking on TREC Web Track documents with the similar queries
• The relevance performance can be worse or better than original queries
• Conducting query transformations on the similar queries to improve the relevance performance
## Query Reformulation using QRU-1

<table>
<thead>
<tr>
<th>SD</th>
<th>MAP</th>
<th>NDCG@20</th>
<th>ERR@20</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline metric</td>
<td>19.13</td>
<td>20.19</td>
<td>8.34</td>
</tr>
<tr>
<td>Best metric</td>
<td>25.00 (+30.7%)</td>
<td>32.88 (+62.9%)</td>
<td>15.09 (+80.9%)</td>
</tr>
<tr>
<td>% outperforming queries</td>
<td>12%</td>
<td>16%</td>
<td>18%</td>
</tr>
<tr>
<td>% topics improved</td>
<td>51%</td>
<td>63%</td>
<td>67%</td>
</tr>
</tbody>
</table>
Query Reformulation using QRU-1

<table>
<thead>
<tr>
<th>SD</th>
<th>MAP</th>
<th>NDCG@20</th>
<th>ERR@20</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline metric</td>
<td>19.13</td>
<td>20.19</td>
<td>8.34</td>
</tr>
<tr>
<td>Best metric</td>
<td>25.00 (+30.7%)</td>
<td>32.88 (+62.9%)</td>
<td>15.09 (+80.9%)</td>
</tr>
<tr>
<td>% outperforming queries</td>
<td>12%</td>
<td>16%</td>
<td>18%</td>
</tr>
<tr>
<td>% topics improved</td>
<td>51%</td>
<td>63%</td>
<td>67%</td>
</tr>
</tbody>
</table>

Only small fraction of query reformulations improve performance
However, for a large number of topics there is at least one good reformulation
<table>
<thead>
<tr>
<th>Topic Title #1</th>
<th>Reformulations</th>
<th>ERR@20</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><em>obama family tree</em></td>
<td></td>
</tr>
<tr>
<td></td>
<td>barack obama ancestry</td>
<td>13.42</td>
</tr>
<tr>
<td></td>
<td>obama s family</td>
<td>32.93</td>
</tr>
<tr>
<td></td>
<td>barack obama s family</td>
<td>32.40</td>
</tr>
<tr>
<td></td>
<td></td>
<td>32.05</td>
</tr>
</tbody>
</table>

**Term Substitution**
Query Expansion

Topic Title #5
Reformulations

<table>
<thead>
<tr>
<th>mitchell college</th>
</tr>
</thead>
<tbody>
<tr>
<td>mitchell college new london</td>
</tr>
<tr>
<td>mitchell college new london ct</td>
</tr>
<tr>
<td>www mitchell edu</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ERR@20</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.2</td>
</tr>
<tr>
<td>19.6</td>
</tr>
<tr>
<td>19.2</td>
</tr>
<tr>
<td>5.7</td>
</tr>
</tbody>
</table>
Abbreviation induction
Take-away Message

• Biggest challenge in search = natural language challenge
• Query document matching
• Latent space model
• QRU-1 dataset
• Open problem: enhancing relevance using QRU-1
• Web search is to deal with the natural language challenge
Thank You!

hangli.hl@huawei.com