Semantic Matching: The Next Big Thing for Natural Language Processing?

Hang Li
Noah’s Ark Lab
Huawei Technologies
Talk Outline

• Semantic Matching
• Web Search
• Question Answering
• Short Text Conversation
• Opportunities and Challenges
Some Well-known NLP Systems

Information Retrieval and Question Answering
- Google Search
- Sogo Question Answering
- IBM Watson
- Apple Siri
- Walframe Alpha
- Microsoft SharePoint Search

Machine Translation
- Google Translate
- Baidu Translation
- Translation Memory: Trados

Text Mining
- SAS TextMiner
- Microsoft SQL Server Text Mining
- Autonomy
Natural Language Analysis (Understanding)

- Pragmatic Analysis
- Semantic Analysis
- Syntactic Analysis
- Lexical Analysis

It is really hot
[Intent: please open the window]

Mary is loved by John
[act: love;agt: John;obj:Mary]

I ate icecream [with a spoon]
I ate icecream [with chocolate]

[南京市][长江大桥]
[南京][市长][江大桥]
Accuracies of Natural Language Analysis

- Lexical Analysis (word segmentation and part-of-speech tagging): practically usable
- Syntactic Analysis: almost usable
- Semantic Analysis: still difficult
- Programmatic Analysis: ?

<table>
<thead>
<tr>
<th></th>
<th>English</th>
<th>Chinese</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pragmatic Analysis</td>
<td>?</td>
<td>?</td>
</tr>
<tr>
<td>Semantic Role Labeling</td>
<td>&gt;=87%</td>
<td>&gt;=75%</td>
</tr>
<tr>
<td>Syntactic Analysis</td>
<td>&gt;=90%</td>
<td>&gt;=80%</td>
</tr>
<tr>
<td>Part of Speech Tagging</td>
<td>&gt;=97%</td>
<td>&gt;=93%</td>
</tr>
<tr>
<td>Word Segmentation</td>
<td>NA</td>
<td>&gt;=95%</td>
</tr>
</tbody>
</table>
Current Approach: Avoid Understanding and Conduct *Matching*
Matching between Two Languages

- **Machine Translation**: A language = source language, B language = target language
- **Question Answering**: A language = question, B language = answer
- **Information Retrieval**: A language = query, B language = document
State of Art Translation Models

Mining word translation, phrase translation, and syntactic translation rules from parallel corpora, generate new translation for given sentence based on the rules

- **Word based Model**
  - Traditional model: IBM Model

- **Phrase based Model**
  - State-of-the-art
  - Google Translate

- **Syntax based Model**
  - Main focus of current research

Alignment is made on parallel corpus
State of Art Information Retrieval Models

Calculating relevance of query with respect to document according to matching degree between query and document

- Term based Model
  - BM25, LM for IR

Intuition: if query words occur many times in document, then document is viewed relevant.
## Biggest Challenge: Mismatch between Query and Document

<table>
<thead>
<tr>
<th>Query</th>
<th>Document</th>
<th>Term Match</th>
<th>Semantic Match</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>
Semantic Matching
(语义匹配)
Next Opportunity: Semantic Matching?

A Language

Semantic Representation

Syntactic Representation

Phrase

Word

B Language

Semantic Representation

Syntactic Representation

Phrase

Word

 Semantic Matching

Syntactic Matching

Phrase Matching

Word Matching

Avoid doing understanding, conducting matching instead
Web Search
Query Document Mismatch is Biggest Challenge in Web Search

NY Times

New York Times

Relation: Relevance
Matching in Latent Space
(Wu et al., WSDM 2013, JMLR 2013)

• 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

$q_1, q_2, q_m, d_1, d_2, d_n$

$L_q, L_d$
Regularized Mapping to Latent Space (RMLS)

- **Setting**
  - Two spaces: \( q \in Q, \ d \in D \)

- **Input**
  - Training data: \( \{(q_i, d_i, c_i)\} \quad c_i \in N^+ \)

- **Output**
  - Matching function \( f(q, d) \)

- **Assumption**
  - Two linear (and sparse) transformations \( L_q, L_d \)
  - Dot product as similarity function \( \langle L_q \cdot q_i, L_d \cdot d_i \rangle \)

- **Optimization**

\[
\arg \max_{L_q, L_d} \sum_{(q_i, d_i)} c_i \langle L_q \cdot q_i, L_d \cdot d_i \rangle \quad \text{st.} \quad |l_q| \leq \theta_q, \ |l_d| \leq \theta_d, \ |l_q| \leq \tau_q, \ |l_d| \leq \tau_d
\]
• Matching between Heterogeneous Data
• Example: Image Annotation
Projecting Keywords and Images into Latent Space
IR Models as Similarity Functions
(Xu and Li 2010)

VSM, BM25, LM, MRF

Mapping functions are diagonal matrices

Query Space

Document Space

New Space
Experimental Results

<table>
<thead>
<tr>
<th>Method</th>
<th>NDCG@1</th>
<th>NDCG@3</th>
<th>NDCG@5</th>
</tr>
</thead>
<tbody>
<tr>
<td>BM25 (baseline)</td>
<td>0.637</td>
<td>0.690</td>
<td>0.690</td>
</tr>
<tr>
<td>SSI</td>
<td>0.538</td>
<td>0.621</td>
<td>0.629</td>
</tr>
<tr>
<td>SVDFeature</td>
<td>0.663</td>
<td>0.720</td>
<td>0.727</td>
</tr>
<tr>
<td>BLTM</td>
<td>0.657</td>
<td>0.702</td>
<td>0.701</td>
</tr>
<tr>
<td>PLS</td>
<td>0.676</td>
<td>0.728</td>
<td>0.736</td>
</tr>
<tr>
<td>RMLIS</td>
<td>0.686</td>
<td>0.732</td>
<td>0.729</td>
</tr>
</tbody>
</table>

- RMLIS and PLS work better than BM25, and other baselines
- RMLIS works equally well as PLS, with higher learning efficiency and scalability
Question Answering
Key: Matching between Question and Answer

Distance between Sun and Earth is ...

Relation: Question-Answer

How far is Sun from Earth?
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-wriring 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

+1/-1?

Prediction System

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

• Training data

\[ ((s_1,t_1), y_1) \cdots ((s_m, t_m), y_m) \]

• Model

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

• 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 Camber was written by Cao Xueqin

Cao Xueqin wrote Dream of the Red Chamber

Hamlet was written by Shakespeare
Re-Writing Rule (2)

• 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)

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

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

\[ \phi_r(s, t) = n \lambda^i \quad \lambda \in (0,1] \]

\[ K(((s_1, t_1), (s_2, t_2)) = \sum_r \phi_r(s_1, t_1) \phi_r(s_2, t_2) \]
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

\[ \langle r_1 = 0, \ldots, r_{100} = 1, \ldots, r_{1001} = 1, \ldots \rangle \quad \langle r_1 = 0, \ldots, r_{100} = 1, \ldots, r_{1001} = 0, \ldots \rangle \]
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 ?
Reformulation of kb-SRK

**Definition**

\[
K((s_1, t_1), (s_2, t_2)) = \sum_r \phi_r(s_1, t_1) \phi_r(s_2, t_2)
\]

\[
\phi_r(s, t) = \sum_{\alpha_s \in k-gram(s)} \phi_r(\alpha_s, \alpha_t) = \sum_{\alpha_t \in k-gram(t)} \phi_r(\alpha_s, \alpha_t)
\]

\[
\phi_r(\alpha_s, \alpha_t) = \lambda^i, \text{if } \alpha_s, \alpha_t \text{ match, } \quad \phi_r(\alpha_s, \alpha_t) = 0, \text{otherwise}
\]

**Equivalent formulation based on fact that summations are interchangeable**

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

\[
K_k = \sum_{r \in R} \phi_r(\alpha_{s_1}, \alpha_{t_1}) \phi_r(\alpha_{s_2}, \alpha_{t_2})
\]
Similarity between Re-writings of K-gram Pairs

Two k-gram pairs:

\[ \alpha_{s_1} = \text{abbccbb} \quad \alpha_{s_2} = \text{abcccedd} \]

\[ \alpha_{t_1} = \text{cbcbcb} \quad \alpha_{t_2} = \text{cbccdedd} \]

Applicable to both pairs:

\[ r_{200} \]

\[ \langle r_1 = 0, \ldots, r_{200} = 1, \ldots, r_{2001} = 1, \ldots \rangle \]

Not applicable to second pair:

\[ r_{2001} \]

\[ \langle r_1 = 0, \ldots, r_{200} = 1, \ldots, r_{2001} = 0, \ldots \rangle \]

Only need to consider the rules applicable to both k-gram pairs.
Efficiently Calculating Number of Applicable Rules on Combined k-gram Pairs

two k-gram pairs

\[ \alpha_{s_1} = \text{abbccbb} \quad \alpha_{s_2} = \text{abccedd} \]

\[ \alpha_{t_1} = \text{cbcbbcb} \quad \alpha_{t_2} = \text{cbccdcdb} \]

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

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

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

identical doubles can be either or not substituted

non-identical doubles must be substituted

\[ \overline{K}_k = \lambda^2 (1 + \lambda^2)(2\lambda^4)(1 + 6\lambda^2 + 6\lambda^4) \]
Time Complexity of Calculation of kb-SRK

- Inner loop
  - Time complexity \( O(k) \)

- Outer loop
  - Empirical time complexity \( O(l^2 \cdot k) \)

\[
l = \max(|s_1|, |t_1|, |s_2|, |t_2|)
\]
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>Lintean 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>

Lexical-based
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>

Lexical-based
Short Text Conversation
Our paper entitled learning to rank has been accepted by ACL.

We are lucky. Our paper has been accepted by SIGIR this year. We are going to present it.

The PC of WSDM noticed us that our paper has been accepted.

Congratulations! It is a great achievement

Great news! Please accept my congrats!

Awesome! It is a great achievement
System of Short Text Conversation

- Given post, find most suitable response
- Large repository of post-response pairs
- Take it as search problem
Learning to Match for Short Text Conversation
Deep Matching Model
(Lu & Li, NIPS 2013)

- Take interactions between texts as input
- Learning features in different granularities using LDA
- Learning parameters using back-propagation
Representing Posts and Comments as Bags of Words

Post

Mats catch mice
Mats chase mice

Comment

Great mats
Poor mice

Bag of words

P_mats, P_catch, P_mice, C_great, C_mats
P_mats, P_chase, P_mice, C_poor, C_mice

Words in post and comment are viewed as different words
Constructing Topics Using Latent Dirichlet Allocation

Bag of words

P_mats, P_catch, P_mice, C_great, C_mats

P_mats, P_chase, P_mice, C_poor, C_mice

Topic

P_mats
P_mice
....

C_mice
C_mats

P_chase
P_catch
....

C_great
C_poor
....

Topics in different granularities form different hidden layers
Construct Neural Network Using Heuristics

Four paired words in pink are connected to a hidden node

Four paired words in blue are connected to another hidden node
Architecture of Deep Matching Model

Examples

Local Model 1: (特产, 土产, 味道, …) || (豆腐, 烤鸭, 甜, 野味, 糯米…)

Local Model 2: (路程, 安排, 地点, …) || (距离, 安全, 隧道, 高速, 机票…)

Q, A interaction space

multiple layers

局部匹配模型

烤鸭啊，想吃热乎的去烤鸭店，如全聚德，真空包装的超市就有

北京有什么出名的特产?
Experimental Results

- Millions of post-response pairs
- For about 60% of posts suitable responses can be found
- Deep matching model works better than linear model

<table>
<thead>
<tr>
<th>Model</th>
<th>MAP</th>
<th>P@1</th>
</tr>
</thead>
<tbody>
<tr>
<td>P2R</td>
<td>0.565</td>
<td>0.489</td>
</tr>
<tr>
<td>P2R + P2P</td>
<td>0.621</td>
<td>0.567</td>
</tr>
<tr>
<td>P2R + MATCH</td>
<td>0.575</td>
<td>0.513</td>
</tr>
<tr>
<td>P2R + P2P + MATCH</td>
<td>0.621</td>
<td>0.574</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Question-Answer</th>
<th>Weibo-Response</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>nDCG@1</td>
<td>nDCG@6</td>
</tr>
<tr>
<td>RANDOM GUESS</td>
<td>0.167</td>
<td>0.550</td>
</tr>
<tr>
<td>PLS</td>
<td>0.285</td>
<td>0.662</td>
</tr>
<tr>
<td>RMLS</td>
<td>0.282</td>
<td>0.659</td>
</tr>
<tr>
<td>SIAMESE NETWORK</td>
<td>0.357</td>
<td>0.735</td>
</tr>
<tr>
<td>DEEPMATCH</td>
<td><strong>0.723</strong></td>
<td><strong>0.856</strong></td>
</tr>
</tbody>
</table>
Opportunities and Challenges
Opportunities and Challenges in Learning to Match

- Data is sparse
- Scale is very large (optimization and computation)
- Data structure is complicated
- Rich semantics needs to be represented (deep models)
- Additional knowledge is available
Thank You!

Contact: hangli.hl@huawei.com