Multi-Perspective Context Modeling for Information Retrieval, Textual Similarity Modeling and NLG

Jinfeng Rao (饶锦锋)
raojinfeng@gmail.com
Facebook Conversational AI
Outline

1. Textual Semantic Modeling
   1.1 Multi-perspective relevance matching
   1.2 Bridging the gap of relevance and semantic matching
   1.3 Quaternion Transformer

2. Temporal Context Modeling
   2.1 Query trend vs. pseudo trend
   2.2 Session modeling with multi-task learning

3. Structural Modeling for NLG
   3.1 Constrained Decoding
   3.2 Tree-to-sequence Model
Related Papers in Part 1

1. J. Rao, L. Liu, J. Lin, Bridging the Gap between Relevance Matching and Semantic Matching with Hierarchical Co-Attention Networks, EMNLP 2019

© Jinfeng Rao 2019
Query

University of Maryland

Document

University of Maryland

UMD, College Park

Maryland

Search Engine

Document Ranking

d1: -2.78

d2: -4.26

d3: -12.5

© Jinfeng Rao 2019
Information Retrieval (state-of-the-art)

**Representation**

- Document is represented by individual words.

- Feature engineering:
  - text match
  - query/doc statistics
  - click…

**Ranking Model**

- **Unsupervised**
  - Language Model
  - VSM
  - TF-IDF
  - BM25
  - DPH
  - COOR

- **Learning to Rank**
  - Pointwise: Logistic Regression, SVM, …
  - Pairwise: RankSVM, LambdaMART…
  - Listwise: ListNet…

- **Neural Network:** DSSM, DRMM, DUET, …

© Jinfeng Rao 2019
Representation-based Matching (DSSM)

Learning Deep Structured Semantic Models for Web Search using Clickthrough Data, Huang et al. CIKM’13

© Jinfeng Rao 2019
Interaction-based Matching (DRMM)

Key consideration:

- Exact match vs. soft match
- Query term weighting
- Diverse matching

A Deep Relevance Matching Model for Ad-hoc Retrieval, Guo et al. CIKM'16
Social Media Search

• Task: Ad-hoc retrieval

• Domain: Microblogs, like Twitter

• Task Description:

  • Given an query q at time t, return a ranked list of all relevant tweets posted before time t.

  • **Query:** BBC world service cuts
  • **Tweet:** BBC news – BBC world service cuts to be outlined to staff
  • **URL:** http://bbc-world-service-to-cut-five-language.html?spref=tw
  • **Hashtag:** #bbcworldservice
Challenges

- Shorter document length, (i.e., 280 characters)
- Informal texts (lots of typos, abbreviations, etc.)
- Heterogeneous relevance signals (URLs, hashtags, etc.)

<table>
<thead>
<tr>
<th>Dataset</th>
<th>TREC 2011</th>
<th>2012</th>
<th>2013</th>
<th>2014</th>
</tr>
</thead>
<tbody>
<tr>
<td># query</td>
<td>49</td>
<td>60</td>
<td>60</td>
<td>55</td>
</tr>
<tr>
<td># query-doc pairs</td>
<td>40K</td>
<td>50K</td>
<td>46K</td>
<td>41K</td>
</tr>
<tr>
<td>% URLs</td>
<td>50%</td>
<td>51%</td>
<td>50%</td>
<td>50%</td>
</tr>
<tr>
<td>% Hashtags</td>
<td>17%</td>
<td>16%</td>
<td>18%</td>
<td>18%</td>
</tr>
<tr>
<td># vocab size</td>
<td>22K</td>
<td>27K</td>
<td>25K</td>
<td>22K</td>
</tr>
<tr>
<td>% OOV</td>
<td>62%</td>
<td>62%</td>
<td>64%</td>
<td>64%</td>
</tr>
</tbody>
</table>
Initial Results of Neural Models

• Task differences result in the poor performance of existing state-of-the-art neural models

<table>
<thead>
<tr>
<th>Method</th>
<th>TREC 2011</th>
<th>2012</th>
<th>2013</th>
<th>2014</th>
</tr>
</thead>
<tbody>
<tr>
<td>QL Baseline</td>
<td>0.358</td>
<td>0.209</td>
<td>0.253</td>
<td>0.392</td>
</tr>
<tr>
<td>DUET [1]</td>
<td>0.174</td>
<td>0.108</td>
<td>0.143</td>
<td>0.257</td>
</tr>
<tr>
<td>DRMM [2]</td>
<td>0.264</td>
<td>0.178</td>
<td>0.210</td>
<td>0.344</td>
</tr>
<tr>
<td>PACRR [3]</td>
<td>0.285</td>
<td>0.205</td>
<td>0.262</td>
<td>0.367</td>
</tr>
</tbody>
</table>

[1] Learning to Match using Local and Distributed Representations of Text for Web Search, Mitra et al. WWW’17
[2] A Deep Relevance Matching Model for Ad-hoc Retrieval, Guo et al. CIKM’16
Our Idea

• Multi-perspective similarity modeling
  • Input-level: query-tweet, query-url
  • Semantic-level: char-level, word-level, phrase-level
  • Weighting-level: multi-level weightings

• Main Components
  • Separate char-level and word-level modeling
  • Hierarchical convolutional layers for long-phrase modeling
  • Pooling-based matching with external weightings
Multi-Perspective Relevance Matching

Word-level Modeling

Character-level Modeling

© Jinfeng Rao 2019
Multi-Perspective Relevance Matching I

• Sentence Encoders
  • Separate encoders for query, document, URL, hashtag
  • Hierarchical conv layers as encoders
  • Given \( h \)-th layer representations \( \mathbf{M} \) of query and document:

\[
\mathbf{M}^h = \text{CNN}^h(\mathbf{M}^{h-1}), h = 1, \ldots, N,
\]

• Similarity Matching
  • Given query representation \( \mathbf{M}_q \) and doc representation \( \mathbf{M}_d \):

\[
\mathbf{S} = \mathbf{M}_q \mathbf{M}_d^T, \mathbf{S} \in \mathbb{R}^{n \times m},
\]

\[
\tilde{S}_{i,j} = \text{softmax}(S_{i,j}) = \frac{e^{S_{i,j}}}{\sum_{k=1}^{m} e^{S_{i,k}}}
\]

\[
\text{Max}(\tilde{S}) = [\max(\tilde{S}_{1,:}), \ldots, \max(\tilde{S}_{n,:})], \text{Max}(\tilde{S}) \in \mathbb{R}^{n};
\]

\[
\text{Mean}(\tilde{S}) = [\text{mean}(\tilde{S}_{1,:}), \ldots, \text{mean}(\tilde{S}_{n,:})], \text{Mean}(\tilde{S}) \in \mathbb{R}^{n};
\]
Multi-Perspective Relevance Matching II

- **External Weighting**
  - Use IDF-based scores to weight each query matching feature.
  - Query term weighting vs. document term weighting
  \[
  \Phi = \{\text{weights}(q) \odot \text{Max}(\tilde{S}), \text{weights}(q) \odot \text{Mean}(\tilde{S})\},
  \]

- **Evidence Integration**
  - MLP layer to combine word-level and character-level matching signals:
  \[
  o = \text{softmax}(\text{MLP}(\Phi^w \sqcup \Phi^c))
  \]

- **Interpolation with Language Model**
  - Linear interpolation of model-based and LM-based score
  \[
  \text{Score}(q, d) = \lambda \cdot \text{NN}(q, d) + (1 - \lambda) \cdot \text{LM}(q, d).
  \]
Experimental Setup

• Datasets: TREC Microblog Track 2011-2014 topic sets

• Experimental condition:
  • three topic sets for training, the left one for testing
  • Randomly selected 10% query set from training set for validation

• Metrics: Mean Average Precision (MAP) and Precision at 30 (P30)

• Baselines:
  • Non-neural baselines:
    • Query Likelihood (QL)
    • Query Expansion (RM3) [11]
    • Learning to rank (LambdaMART) [12]
      • text-based features (5)
      • URL-based features (2)
      • hashtag-based features (2)
  • Neural baselines:
    • PACRR (2017) [19]
    • DRMM (2016) [17]
    • K-NRM (2017) [18]
  • Interpolation baselines:
    • PACRR+QL
    • DRMM+QL
    • K-NRM+QL

© Jinfeng Rao 2019
Main Results

<table>
<thead>
<tr>
<th>Method/Dataset</th>
<th>TREC 2011</th>
<th>2012</th>
<th>2013</th>
<th>2014</th>
</tr>
</thead>
<tbody>
<tr>
<td>Query Likelihood(QL)</td>
<td>0.358</td>
<td>0.209</td>
<td>0.253</td>
<td>0.392</td>
</tr>
<tr>
<td>RM3</td>
<td>0.382</td>
<td>0.234</td>
<td>0.276</td>
<td>0.448</td>
</tr>
<tr>
<td>L2R (text)</td>
<td>0.355</td>
<td>0.207</td>
<td>0.239</td>
<td>0.382</td>
</tr>
<tr>
<td>L2R (text +URL)</td>
<td>0.382</td>
<td>0.232</td>
<td>0.249</td>
<td>0.397</td>
</tr>
</tbody>
</table>

Neural Baselines

<table>
<thead>
<tr>
<th>Method</th>
<th>TREC 2011</th>
<th>2012</th>
<th>2013</th>
<th>2014</th>
</tr>
</thead>
<tbody>
<tr>
<td>K-NRM (2017)</td>
<td>0.252</td>
<td>0.161</td>
<td>0.175</td>
<td>0.347</td>
</tr>
<tr>
<td>PACRR (2017)</td>
<td>0.285</td>
<td>0.205</td>
<td>0.263</td>
<td>0.367</td>
</tr>
</tbody>
</table>

Our Approach

<table>
<thead>
<tr>
<th>Method</th>
<th>TREC 2011</th>
<th>2012</th>
<th>2013</th>
<th>2014</th>
</tr>
</thead>
<tbody>
<tr>
<td>MP-HCNN</td>
<td>0.383</td>
<td>0.234</td>
<td>0.282</td>
<td>0.430</td>
</tr>
<tr>
<td>MP-HCNN+</td>
<td>0.404</td>
<td>0.246</td>
<td>0.290</td>
<td>0.442</td>
</tr>
</tbody>
</table>

- Observations
  - Existing neural approaches suffered, while a simple interpolation works!
  - Our approaches (MP-HCNN) significantly outperformed all competitive baselines.
### Ablation Study

<table>
<thead>
<tr>
<th>Setting/Dataset</th>
<th>TREC 2011</th>
<th>2012</th>
<th>2013</th>
<th>2014</th>
</tr>
</thead>
<tbody>
<tr>
<td>QL Baseline</td>
<td>0.358</td>
<td>0.209</td>
<td>0.253</td>
<td>0.392</td>
</tr>
<tr>
<td><strong>Full Model</strong></td>
<td><strong>0.383</strong></td>
<td><strong>0.234</strong></td>
<td><strong>0.282</strong></td>
<td><strong>0.430</strong></td>
</tr>
</tbody>
</table>

#### Ablation Setting

- **IDF weighting**
  - 0.351* |
  - 0.211* |
  - 0.272* |
  - 0.399* |
- **Max pooling**
  - 0.098* |
  - 0.076* |
  - 0.092* |
  - 0.193* |
- **Mean pooling**
  - 0.369* |
  - 9.225 |
  - 0.276 |
  - 0.390* |
- **URL char part**
  - 0.359* |
  - 0.213* |
  - 0.279* |
  - 0.403* |
- **Doc char part**
  - 0.360* |
  - 0.219* |
  - 0.275* |
  - 0.401* |
- **Doc word part**
  - 0.165* |
  - 0.076* |
  - 0.098* |
  - 0.185* |

* symbol denotes the score is significantly lower than full model at p<0.05.

- **Observations**
  - Word-level modeling and max-pooling similarity are essential.
  - Char-level modeling of documents and URL, with IDF weighting further contributes.
Model Effectiveness Study

Model Effectiveness with #conv layers

Matching Evidence Breakdown of top 5 performing queries

<table>
<thead>
<tr>
<th>Category</th>
<th>Percentage (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exact word match</td>
<td>100</td>
</tr>
<tr>
<td>Exact phrase match</td>
<td>44</td>
</tr>
<tr>
<td>Partial paraphrase match</td>
<td>59</td>
</tr>
<tr>
<td>Partial URL match</td>
<td>29</td>
</tr>
</tbody>
</table>

Per-query difference of model score w.r.t QL baseline

© Jinfeng Rao 2019
## Error Analysis

<table>
<thead>
<tr>
<th>ID</th>
<th>Query</th>
<th>Sample Tweet</th>
<th>Label</th>
<th>Score/Rank</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>QL</td>
</tr>
</tbody>
</table>
| 1  | 2: 2022 fifa soccer | #ps3 best sellers: fifa soccer 11 ps3 #cheaptweet  
https://www.amazon.com/fifa-soccer-11-playstation-3 | I | 7.33(#54) | 0.85(#1) |
| 2  | qatar’s 2022 fifa world cup stadiums:  
https://wordlesstech.com/qatars-2022-fifa-world-cup-stadiums/ | | R | 10.58(#2) | 0.41(#105) |
| 3  | 2022 world cup could be held at end of year: fifa : lausanne switzerland the 2022 world cup in qatar:  
http://www.reuters.com/article/us-soccer-world-blatter | | R | 11.25(#1) | 0.31(#127) |

Figure: error analysis for the worst-performing topic 2. Label I denotes irrelevant, and R denotes relevant. The pink ground color denotes the score of phrase-matching by MP-HCNN. The brighter the color, the higher the matching score.

- **Observations**
  - Our model put more emphasizes on phrase match (such as “fifa soccer”)
  - The OOV word “2022” underweights its contribution.
  - Relevance signals in URLs are important.
Outline

1. Textual Semantic Modeling
   1.1 Multi-perspective relevance matching
   1.2 Bridging the gap of relevance and semantic matching
   1.3 Quaternion Transformer

2. Temporal Context Modeling
   2.1 Query trend vs. pseudo trend
   2.2 Session modeling with multi-task learning

3. Structural Modeling for NLG
   3.1 Constrained Decoding
   3.2 Tree-to-sequence Model
Relevance Matching (IR) vs. Semantic Matching (NLP)

- **Relevance Matching (IR):**
  - Ad-hoc retrieval, short text search (i.e., Twitter)
  - Focus on keyword-based matching, term weighting
  - Unsymmetrical modeling of query (short) and documents (long)

- **Semantic Matching (NLP):**
  - Question answering, paraphrase detection, textual similarity measurement, etc.
  - Focus on semantics understanding, compositional meaning
  - Symmetrical modeling of two sentences

<table>
<thead>
<tr>
<th>Task</th>
<th>Label</th>
<th>Sentence A</th>
<th>Sentence B</th>
</tr>
</thead>
<tbody>
<tr>
<td>RM</td>
<td>1</td>
<td>2022 FIFA soccer</td>
<td>2022 world cup fifa could be held at end of year in Qatar</td>
</tr>
<tr>
<td>SM</td>
<td>0</td>
<td>Does RBI send its employees for higher education, like MBA?</td>
<td>Does EY send its employees for higher education, like MBA?</td>
</tr>
</tbody>
</table>
Architectures Comparison

Relevance Matching

A Deep Relevance Matching Model for Ad-hoc Retrieval, Guo et al. CIKM’16

Semantic Matching

Bidirectional Attention Flow for Machine Comprehension, Seo et al. ICLR 2017

© Jinfeng Rao 2019
Our Method: Hierarchical Co-Attention Network (HCAN)

Three major components:
- Hierarchical Representation Learning
- Weighted Relevance Matching
- Semantic Matching
HCAN Encoders

• Deep Encoder
  • Hierarchical conv layers as encoders
  • Weight sharing and fast for phrase modeling
  • Given $h$-th layer representations $M$ for a text sequence:

  $$M^h = \text{CNN}^h(M^{h-1}), h = 1, \ldots, N,$$

• Wide Encoder
  • Parallel conv layers (total $N$ layers) with different window sizes $[1, 2, \ldots, N]$
  • Allows to apply ngram weightings explicitly

• Contextual Encoder
  • Hierarchical BiLSTM layers to model long-range contextual dependencies
  • Hard to apply term weighting and is slower in training and inference

© Jinfeng Rao 2019
**HCAN Relevance and Semantic Matching**

- **Relevance Matching**
  - Hierarchical BiLSTM layers to model long-range contextual dependencies
  - Hard to apply term weighting and is slower in training and inference

\[
S = U_q U_c^T, \quad S \in \mathbb{R}^{n \times m}, \quad \tilde{S}_{i,j} = \text{softmax}(S_{i,j}) = \frac{e^{S_{i,j}}}{\sum_{k=1}^{m} e^{S_{i,k}}}. 
\]

\[
\Phi = \{\text{weights}(q) \odot \text{Max}(\tilde{S}), \text{weights}(q) \odot \text{Mean}(\tilde{S})\},
\]

- **Semantic Matching**
  - Cross-attention semantic modeling

\[
A = \text{REP}(U_q W_q) + \text{REP}(U_c W_c) + U_q W_b U_c^T
\]

\[
\tilde{U}_q = A^T U_q
\]

\[
\tilde{U}_c = \text{REP}(\max_{\text{col}}(A) U_c)
\]

\[
\tilde{U}_q \in \mathbb{R}^{m \times F}, \quad \tilde{U}_c \in \mathbb{R}^{m \times F}
\]

- **Contextual integration**

\[
H = [U_c; \tilde{U}_q; U_c \otimes \tilde{U}_q; \tilde{U}_c \otimes \tilde{U}_q]
\]

\[
O_{SM} = \text{BiLSTM}(H)
\]

\[
H \in \mathbb{R}^{m \times 4F}, \quad O_{SM} \in \mathbb{R}^d
\]

© Jinfeng Rao 2019
Experiments

- Three NLP tasks
  - Answer Sentence Selection (TrecQA)
  - Paraphrase Detection (TwitterURL)
  - Semantic Textual Similarity (Quora)

- Two IR datasets
  - TREC Microblog 2013
  - TREC Microblog 2014

- Baselines
  - InferSent (Conneau et al. 2017)
  - ESIM (Chen et al. 2017)
  - DecAtt (Parikh et al. 2017)
  - PWIM (He et al. 2016)
## Major Results

### Semantic Matching Tasks

<table>
<thead>
<tr>
<th>Model</th>
<th>TrecQA MAP</th>
<th>TrecQA MRR</th>
<th>TwitterURL F1</th>
<th>Quora Acc</th>
</tr>
</thead>
<tbody>
<tr>
<td>InferSent</td>
<td>0.521</td>
<td>0.559</td>
<td>0.746</td>
<td>0.866</td>
</tr>
<tr>
<td>DecAtt</td>
<td>0.660</td>
<td>0.712</td>
<td>0.652</td>
<td>0.845</td>
</tr>
<tr>
<td>ESIM&lt;sub&gt;seq&lt;/sub&gt;</td>
<td>0.771</td>
<td>0.795</td>
<td>0.748</td>
<td>0.850</td>
</tr>
<tr>
<td>ESIM&lt;sub&gt;tree&lt;/sub&gt;</td>
<td>0.698</td>
<td>0.734</td>
<td>0.740</td>
<td>0.755</td>
</tr>
<tr>
<td>ESIM&lt;sub&gt;seq+tree&lt;/sub&gt;</td>
<td>0.749</td>
<td>0.768</td>
<td>0.759</td>
<td>0.854</td>
</tr>
</tbody>
</table>

**State-of-the-art**

- Rao et al. (2016): 0.780, 0.834
- He and Lin (2016): 0.739, 0.795
- Gong et al. (2018): -

**Our Approach**

- RM: 0.756, 0.812
- SM: 0.663, 0.725
- HCAN: 0.774, **0.843**

### Relevance Matching Tasks

<table>
<thead>
<tr>
<th>Model</th>
<th>TREC-2013 MAP</th>
<th>TREC-2014 MAP</th>
<th>TREC-2014 P@30</th>
</tr>
</thead>
<tbody>
<tr>
<td>QL</td>
<td>0.2532</td>
<td>0.3924</td>
<td>0.6182</td>
</tr>
<tr>
<td>RM3</td>
<td>0.2766</td>
<td><strong>0.4480</strong></td>
<td>0.6339</td>
</tr>
<tr>
<td>L2R</td>
<td>0.2477</td>
<td>0.3943</td>
<td>0.6200</td>
</tr>
</tbody>
</table>

**Neural Baselines**

- DUET: 0.1380, 0.2528
- DRMM: 0.2102, 0.4061
- K-NRM: 0.1750, 0.3178
- PACRR: 0.2627, 0.4872

**Our HCAN Approach**

- RM: 0.2818, 0.5222
- SM: 0.1365, 0.2411
- HCAN: **0.2920**, **0.5328**

### Observations

- RM is strong across all datasets, showing keyword-based matching is helpful for SM tasks as well.
- HCAN is consistently better than RM and SM.
## Ablation Study

### Encoder Comparison

<table>
<thead>
<tr>
<th>Encoders</th>
<th>Matching</th>
<th>TrecQA MAP</th>
<th>TrecQA MRR</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Deep Encoder</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RM</td>
<td>0.756</td>
<td>0.812</td>
<td></td>
</tr>
<tr>
<td>SM</td>
<td>0.663</td>
<td>0.725</td>
<td></td>
</tr>
<tr>
<td>comb</td>
<td>0.774</td>
<td>0.843</td>
<td></td>
</tr>
<tr>
<td><strong>Wide Encoder</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RM</td>
<td>0.758</td>
<td>0.806</td>
<td></td>
</tr>
<tr>
<td>SM</td>
<td>0.673</td>
<td>0.727</td>
<td></td>
</tr>
<tr>
<td>comb</td>
<td>0.770</td>
<td>0.847</td>
<td></td>
</tr>
<tr>
<td><strong>Contextual Encoder</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RM</td>
<td>0.690</td>
<td>0.736</td>
<td></td>
</tr>
<tr>
<td>SM</td>
<td>0.668</td>
<td>0.735</td>
<td></td>
</tr>
<tr>
<td>comb</td>
<td>0.739</td>
<td>0.790</td>
<td></td>
</tr>
</tbody>
</table>

© Jinfeng Rao 2019
Learning Efficiency

**Validation Losses**

- Observations
  - HCAN is learning much faster and more data-efficient.

© Jinfeng Rao 2019
## Qualitative Example

<table>
<thead>
<tr>
<th>Label</th>
<th>SM Score</th>
<th>RM Score</th>
<th>HCAN Score</th>
<th>Sample Pair</th>
<th></th>
</tr>
</thead>
</table>
| 1     | 1, 0.9119| 0, 0.9353| 1, 0.5496 | - How does it feel to kill a human?  
- How does it feel to be a murderer? |  |
| 1     | 0, 0.9689| 1, 0.8762| 1, 0.8481 | - What are the time dilation effects on the ISS?  
- According to the theory of relativity, time runs slowly under the influence of gravity. Is there any time dilation experienced on the ISS? |  |
| 0     | 0, 0.9927| 1, 0.8473| 1, 0.7280 | - Does RBI send its employees for higher education such as MBA, like sponsoring the education or allowing paid/unpaid leaves?  
- Does EY send its employees for higher education such as MBA, like sponsoring the education or allowing paid/unpaid leaves? |  |

Table 2: Sample pairs on Quora dataset. Phrases with large attention weights are highlighted in orange and red.
Outline

1. Textual Semantic Modeling
   1.1 Multi-perspective relevance matching
   1.2 Bridging the gap of relevance and semantic matching
   1.3 Quaternion Transformer

2. Temporal Context Modeling
   2.1 Query trend vs. pseudo trend
   2.2 Session modeling with multi-task learning

3. Structural Modeling for NLG
   3.1 Constrained Decoding
   3.2 Tree-to-sequence Model
Background: Transformer

Transformer Arch

Multi-head Attention

\[
\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V
\]

\[
\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \ldots, \text{head}_h)W^O
\]

where \( \text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V) \)
Over-Parameterized Neural Networks

<table>
<thead>
<tr>
<th>Model</th>
<th>#Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>BERT (base)</td>
<td>110M</td>
</tr>
<tr>
<td>BERT (large)</td>
<td>340M</td>
</tr>
<tr>
<td>GPT</td>
<td>117M</td>
</tr>
<tr>
<td>GPT-2</td>
<td>1542M</td>
</tr>
</tbody>
</table>

- **Our Solution:**
  - Quaternion Networks

**Consumption**

<table>
<thead>
<tr>
<th>Activity</th>
<th>CO₂e (lbs)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Air travel, 1 passenger, NY↔SF</td>
<td>1984</td>
</tr>
<tr>
<td>Human life, avg, 1 year</td>
<td>11,023</td>
</tr>
<tr>
<td>American life, avg, 1 year</td>
<td>36,156</td>
</tr>
<tr>
<td>Car, avg incl. fuel, 1 lifetime</td>
<td>126,000</td>
</tr>
</tbody>
</table>

**Training one model (GPU)**

<table>
<thead>
<tr>
<th>Model</th>
<th>CPU (Gbps)</th>
</tr>
</thead>
<tbody>
<tr>
<td>NLP pipeline (parsing, SRL)</td>
<td>39</td>
</tr>
<tr>
<td>w/ tuning &amp; experimentation</td>
<td>78,468</td>
</tr>
<tr>
<td>Transformer (big)</td>
<td>192</td>
</tr>
<tr>
<td>w/ neural architecture search</td>
<td>626,155</td>
</tr>
</tbody>
</table>

Energy and Policy Considerations for Deep Learning in NLP, Strubell et al. ACL 2019

© Jinfeng Rao 2019
Quaternion Basics

- Quaternion
  - Definition: a quaternion contains three imaginary components.
    \[ Q = r + xi + yj + zk, \text{ where } ijk = i^2 = j^2 = k^2 = -1 \]
  - Addition: \[ Q + P = Q_r + P_r + (Q_x + P_x)i + (Q_y + P_y)j + (Q_z + P_z)k, \]
  - Norm: \[ Q^d = \frac{Q}{\sqrt{r^2 + x^2 + y^2 + z^2}}. \]
  - Multiplication (Hamilton Product):
    \[ Q \otimes P = (Q_rP_r - Q_xP_x - Q_yP_y - Q_zP_z) + (Q_xP_r + Q_rP_x - Q_zP_y + Q_yP_z)i + (Q_yP_r + Q_zP_x + Q_rP_y - Q_xP_z)j + (Q_zP_r - Q_yP_x + Q_xP_y + Q_rP_z)k, \]
How Quaternion Saves Parameters?

- Quaternion Feed Forward Layer (QFFN)
  - Given $W$ as Quaternion layer weight and $Q$ as layer input, $W \otimes Q$:
    
    $\begin{bmatrix}
    W_r & -W_x & -W_y & -W_z \\
    W_x & W_r & -W_z & W_y \\
    W_y & W_z & W_r & -W_x \\
    W_z & -W_y & W_x & W_r \\
    \end{bmatrix}
    \begin{bmatrix}
    r \\
    x \\
    y \\
    z \\
    \end{bmatrix}$

    where $W = W_r + W_x i + W_y j + W_z k$

    75% Parameter Reduction!

Figure 1: 4 weight parameter variables $(W_r, W_x, W_y, W_z)$ are used in 16 pairwise connections between components of the input and output Quaternions.
**Quaternion Attention**

- **Aligned Representations with Attention**
  
  Let $A \in \mathbb{H}^{l_a \times d}$ and $B \in \mathbb{H}^{l_b \times d}$
  
  $$E = A \otimes B^\top, \quad E \in \mathbb{H}^{l_a \times l_b}.$$  

  - Obtain aligned representations $A'$ and $B'$:
    
    $$G = \text{ComponentSoftmax}(E') \quad \text{and} \quad F = \text{ComponentSoftmax}(E^\top)$$
    
    $$B' = G_R B_R + G_X B_X i + G_Y B_Y j + G_Z B_Z k,$$
    
    $$A' = F_R A_R + F_X A_X i + F_Y A_Y j + F_Z A_Z k,$$

  - Compare and Aggregate:
    
    $$C_1 = \sum \text{QFFN}([A'_i; B_i, A'_i \otimes B_i; A'_i - B_i])$$
    
    $$C_2 = \sum \text{QFFN}([B'_i; A_i, B'_i \otimes A_i; B'_i - A_i]),$$
    
    $$Y = \text{QFFN}([C_1; C_2; C_1 \otimes C_2; C_1 - C_2]).$$
Quatetion Transformer

- Quaternion Self-Attention

\[ A = \text{ComponentSoftmax}(\frac{Q \otimes K}{\sqrt{d_k}})V. \]

\[ Q = W_q \otimes X; K = W_k \otimes X; V = W_v \otimes X, \]

- Quaternion Embedding/Loss
  - Quaternion conversion of pre-trained real embeddings
  - A linear combination for cross-entropy loss calculation

\[ Y = \text{Softmax}(W([r; x; y; z]) + b), \]

- Two Configurations
  - Quaternion (full): all quaternion operation
  - Quaternion (partial): only using quaternion-based self-attention
## Quaternion Experiments

### Experiments on Pairwise Classification/Ranking Tasks

<table>
<thead>
<tr>
<th>Task</th>
<th>NLI</th>
<th>QA</th>
<th>Paraphrase</th>
<th>RS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Measure</td>
<td>Accuracy</td>
<td>MAP/MRR</td>
<td>Accuracy</td>
<td>Top-1</td>
</tr>
<tr>
<td>Model</td>
<td>SNLI</td>
<td>SciTail</td>
<td>MNLI</td>
<td>WikiQA</td>
</tr>
<tr>
<td>DeAtt ($d = 50$)</td>
<td>83.4</td>
<td>73.8</td>
<td>69.9/70.9</td>
<td>66.0/67.1</td>
</tr>
<tr>
<td>DeAtt ($d = 200$)</td>
<td><strong>86.2</strong></td>
<td>79.0</td>
<td><strong>73.6/73.9</strong></td>
<td><strong>67.2/68.3</strong></td>
</tr>
<tr>
<td>Q-Att ($d = 50$)</td>
<td>85.4</td>
<td><strong>79.6</strong></td>
<td>72.3/72.9</td>
<td>66.2/68.1</td>
</tr>
</tbody>
</table>

### Experiments on Sentiment Classification Tasks

<table>
<thead>
<tr>
<th>Model</th>
<th>IMDb</th>
<th>SST</th>
<th># Params</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transformer</td>
<td><strong>82.6</strong></td>
<td>78.9</td>
<td>400K</td>
</tr>
<tr>
<td>Quaternion Transformer (full)</td>
<td><strong>83.9 (+1.3%)</strong></td>
<td>80.5 (+1.6%)</td>
<td>100K (-75.0%)</td>
</tr>
<tr>
<td>Quaternion Transformer (partial)</td>
<td>83.6 (+1.0%)</td>
<td><strong>81.4 (+2.5%)</strong></td>
<td>300K (-25.0%)</td>
</tr>
</tbody>
</table>
## Quaternion Experiments II

### Experiments on Machine Translation Tasks

<table>
<thead>
<tr>
<th>Model</th>
<th>IWSLT’15 En-Vi</th>
<th>WMT’16 En-Ro</th>
<th>WMT’18 En-Et</th>
<th># Params</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transformer Base</td>
<td>28.4</td>
<td><strong>22.8</strong></td>
<td>14.1</td>
<td>44M</td>
</tr>
<tr>
<td>Quaternion Transformer (full)</td>
<td>28.0</td>
<td>18.5</td>
<td>13.1</td>
<td>11M (-75%)</td>
</tr>
<tr>
<td>Quaternion Transformer (partial)</td>
<td><strong>30.9</strong></td>
<td>22.7</td>
<td><strong>14.2</strong></td>
<td>29M (-32%)</td>
</tr>
</tbody>
</table>

### Experiments on Mathematic Language Understanding Tasks

<table>
<thead>
<tr>
<th>Model</th>
<th>Acc / Seq</th>
<th># Params</th>
</tr>
</thead>
<tbody>
<tr>
<td>Universal Transformer</td>
<td>78.8</td>
<td>-</td>
</tr>
<tr>
<td>ACT U-Transformer</td>
<td>84.9</td>
<td>-</td>
</tr>
<tr>
<td>Transformer</td>
<td>76.1</td>
<td>400K</td>
</tr>
<tr>
<td>Quaternion Transformer (full)</td>
<td><strong>78.9 (+2.8%)</strong></td>
<td>100K (-75%)</td>
</tr>
<tr>
<td>Quaternion Transformer (partial)</td>
<td><strong>84.4 (+8.3%)</strong></td>
<td>300K (-25%)</td>
</tr>
</tbody>
</table>
### NLP Gaints

- **BERT**, Devlin et al. NAACL’19
  - Transformer model, pretrained on 112GB texts
  - Masked Language Model and Next Sentence Prediction Task

- **SpanBERT**, Chen et al. ACL’19
  - Dynamically sample spans for MLM and add span-based loss in pretraining
  - Next Sentence Prediction loss removed

- **XLNet**, Yang et al. 2019
  - Non-autoregressive formulation of language modeling
  - Fix the pretrain-finetune discrepancy introduced by the mask mechanism

- **ERINE 2.0**, Sun et al. 2019
  - Phrase-based and knowledge-based masking mechanism
  - Multi-task learning for pretraining (word-level, structure-level, semantic-level)
Stand on Shoulders of Giants

• BERT, Devlin et al. NAACL’19
  • Transformer model, pretrained on 112GB texts
  • Masked Language Model and Next Sentence Prediction Task

• SpanBERT, Chen et al. ACL’19
  • Dynamically sample spans for MLM and add span-based loss in pretraining
  • Next Sentence Prediction Loss is removed

• XLNet, Yang et al. 2019
  • Non-autoregressive formulation of language modeling
  • Fix the pretrain-finetune discrepancy introduced by the mask mechanism

• ERINE 2.0, Sun et al. 2019
  • Phrase-based and knowledge-based masking mechanism
  • Multi-task learning for pretraining (word-level, structure-level, semantic-level)
Outline

1. Textual Semantic Modeling
   1.1 Multi-perspective relevance matching
   1.2 Bridging the gap of relevance and semantic matching
   1.3 Quaternion Transformer

2. Temporal Context Modeling
   2.1 Query trend vs. pseudo trend
   2.2 Session modeling with multi-task learning

3. Structural Modeling for NLG
   3.1 Constrained Decoding
   3.2 Tree-to-sequence Model
Related Papers in Part 2


Motivation

• Distribution of relevant documents in time
  • horizontal axis denotes time prior to query time
  • green for relevant document, red for highly relevant, gray for irrelevant

• Insights
  • relevant docs cluster in time
  • clustering patterns vary across queries

© Jinfeng Rao 2019
Background: Combining Lexical & Temporal Evidence

- Within the language modeling framework
  - Recency Prior (Li and Croft et al.)
    \[ P(D|Q) \propto P(Q|D)P(D) \]
  - Independent Evidence (Dakka et al.)
    \[ P(D|Q) = P(W_D, T_D|Q) = P(W_D|Q)P(T_D|Q) \]
- Log-linear Combination
  \[ P(R|D, Q) = P(R|W_D, Q)^{1-\alpha} \cdot P(R|T_D, Q)\alpha \]
  \[
  \alpha \in [0, 1]
  \]
Background: Kernel Density Estimation

• A two-stage process for pseudo trend estimation:
  • Retrieve docs to estimate the ground truth distribution (pseudo trend)
  • Rerank docs with the estimated pseudo trend.

• **Kernel density estimation**, Efron et al. SIGIR’14 [10]

\[
\hat{f}_\omega(x) = \frac{1}{nh} \sum_{i=0}^{n} \omega_i K \left( \frac{x - x_i}{h} \right)
\]

Four weighting schemas w:
1. Uniform
2. Score-based
3. Rank-based
4. Oracle (upper bound)
Query Trend Motivation

• Research question: can we make use of the temporal statistics of query terms (query trends) to predict the ground truth?

• What is query trend? E.g., collection frequencies of query terms for each 5 minutes.

• An example of ground truth and term trends for query MB127 “hagel nomination filibustered” from TREC 2013 topic set.
Regression of Query Trends

**Goal:** Approximate the ground truth ($Y$) by taking a weighted sum of all query trends ($f_t$).

$$Y \approx \sum_{t} w_t f_t$$
Term Importance Modeling

• Bursty terms can be more informative.

• We adopt entropy definition to measure the importance of terms.

• Given the counts of a particular term \( t \) (unigram/bigram) \( \{c_1, c_2, \ldots, c_n\} \),

\[
\text{Entropy}(t) = - \sum_i \frac{c_i}{C} \log \frac{c_i}{C}
\]

lower entropy = bursty term trend = more important
Two questions in this non-linear regression modeling:
- Q1: How to model the weights of different query terms?
- Q2: How to differentiate the contribution from unigrams with bigrams?

Q1 solution: exponential mapping from entropy to term weight
\[ w_t = \exp(\theta \cdot e_t) - 1 \]

Q2 solution: assume unigram weight \( u_i \), then bigram weight \( (1-u_i) \)
\[ u_i = \text{logistic}(R_i, \gamma) = \frac{1}{1 + \exp(-\gamma R_i)} \]

where \( R_i \) is the difference between the maximum unigram entropy and maximum bigram entropy.

Intuition: \( R_i > 0 \Rightarrow \max(\text{unigram_entropy}) > \max(\text{bigram_entropy}) \Rightarrow u_i > 0.5 \)
Regression of Query Trends

- Problem reformulation:

\[
Y \approx \sum_t w_t f_t \quad \Rightarrow \quad Y_i \approx u_i U_i w_i^u + (1 - u_i) B_i w_i^b
\]

- Objective Loss:

\[
L = \sum_{i=1}^{N_q} \|Y_i - (u_i U_i (e^{\alpha E_i^u} - 1)^T + (1 - u_i) B_i (e^{\beta E_i^b} - 1)^T)\|^2 \\
+ \lambda (\alpha^2 + \beta^2 + \gamma^2)
\]

which can be solved with gradient descent algorithm (more details in paper [4]).
Combine Query Trend with Pseudo Trend

- Two ways to estimate the ground truth distribution:
  - **Document-level:** pseudo trend through an initial retrieval
  - **Term-level:** regression over query trends

- Combine query trend and pseudo trend in a linear ranking model:

\[
S_d = \sum_i \alpha_i \cdot F_i(d, q) \quad \text{s.t.} \quad \sum_i \alpha_i = 1.
\]
Experimental Setup

• Dataset: TREC Microblog Track 2013-2014, total 115 topics.

• Experimental conditions:
  • Odd-even: odd numbered topics (57 topics) for training, even (58 topics) for testing
  • Even-odd: switch train/test split
  • Cross: 4-fold cross validation

• Baselines:
  • Query Likelihood (QL)
  • Recency Prior, Li et al. CIKM’03 [8]
  • Moving Window, Dakka et al. TKDE’12 [9]
  • Kernel Density Estimation (KDE), SIGIR’14 [10]
    • Uniform (IRDu)
    • Score-based (IRDs)
    • Rank-based (IRDr)
    • Oracle (upper bound)
Main Results

<table>
<thead>
<tr>
<th>ID</th>
<th>Method</th>
<th>Odd-Even AP</th>
<th>Odd-Even P30</th>
<th>Even-Odd AP</th>
<th>Even-Odd P30</th>
<th>Cross AP</th>
<th>Cross P30</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Query Likelihood (QL) [15]</td>
<td>0.271</td>
<td>0.475</td>
<td>0.357</td>
<td>0.564</td>
<td>0.315</td>
<td>0.520</td>
</tr>
<tr>
<td>2</td>
<td>Recency prior [12]</td>
<td>0.277</td>
<td>0.499</td>
<td>0.359</td>
<td>0.574</td>
<td>0.313</td>
<td>0.534</td>
</tr>
<tr>
<td>3</td>
<td>Moving Window (WIN) [5]</td>
<td>0.283</td>
<td>0.487</td>
<td>0.358</td>
<td>0.567</td>
<td>0.319</td>
<td>0.527</td>
</tr>
<tr>
<td>4</td>
<td>KDE [8]</td>
<td>IRDu</td>
<td>0.273</td>
<td>0.481</td>
<td>0.350</td>
<td>0.566</td>
<td>0.308</td>
</tr>
<tr>
<td></td>
<td>IRDs</td>
<td>0.274</td>
<td>0.487</td>
<td>0.353</td>
<td>0.577</td>
<td>0.314</td>
<td>0.530</td>
</tr>
<tr>
<td></td>
<td>IRDr</td>
<td>0.288$^{1,4,5}$</td>
<td>0.517$^{1,3,4,5}$</td>
<td>0.360</td>
<td>0.588$^{1,2,3,4}$</td>
<td>0.327$^{1,2,4,5}$</td>
<td>0.552$^{1,2,3,4,5}$</td>
</tr>
<tr>
<td>7</td>
<td>This work</td>
<td>QT</td>
<td>0.278</td>
<td>0.492$^{1,4}$</td>
<td>0.367$^{1,4,5}$</td>
<td>0.587$^{1,2,3,4}$</td>
<td>0.320</td>
</tr>
<tr>
<td></td>
<td>Reg</td>
<td>0.276</td>
<td>0.488$^{1}$</td>
<td>0.366$^{1,4,5}$</td>
<td>0.576$^{1}$</td>
<td>0.329$^{1,2,4,5}$</td>
<td>0.535$^{1,4}$</td>
</tr>
<tr>
<td></td>
<td>QT-IRD$^r$</td>
<td>0.290$^{1,2,4,5}$</td>
<td>0.522$^{1,2,3,4,5}$</td>
<td>0.368$^{1,2,3,4}$</td>
<td>0.596$^{1,2,3,4,5}$</td>
<td>0.328$^{1,2,4,5}$</td>
<td>0.565$^{1,2,3,4,5}$</td>
</tr>
<tr>
<td>10</td>
<td>This work</td>
<td>Reg-IRD$^r$</td>
<td>0.302$^{1,2,3,4,5,6}$</td>
<td>0.535$^{1,2,3,4,5}$</td>
<td>0.368$^{1,2,3,4}$</td>
<td>0.596$^{1,2,3,4,5}$</td>
<td>0.332$^{1,2,3,4,5}$</td>
</tr>
<tr>
<td>11</td>
<td>Oracle</td>
<td></td>
<td></td>
<td>0.382$^{1,2,3,4,5,6}$</td>
<td>0.636$^{1,2,3,4,5,6}$</td>
<td>0.349$^{1,2,3,4,5,6}$</td>
<td>0.586$^{1,2,3,4,5,6}$</td>
</tr>
</tbody>
</table>

- Conclusions:
  - KDE with rank-based weights (IRD$^r$) is the strongest baseline.
  - Our approach (Reg-IRD$^r$) significantly outperforms all baselines, and is even close to the upper bound in some splits.
Per-topic P30 improvement against the Query Likelihood (QL) and the best KDE baseline (IRDr).
Analysis of the Best-Performing Topic 144

• How query trend signals help?
  • red color for ground truth distribution
  • green for pseudo trend estimated by the best KDE method (IRDr)
  • blue for query trends.

• Conclusion: A combination of pseudo trend (KDE) and query trend (Our approaches) provides a more accurate estimation to the ground truth distribution.
Outline

1. Textual Semantic Modeling
   1.1 Multi-perspective relevance matching
   1.2 Bridging the gap of relevance and semantic matching
   1.3 Quaternion Transformer

2. Temporal Context Modeling
   2.1 Query trend vs. pseudo trend
   2.2 Session modeling with multi-task learning

3. Structural Modeling for NLG
   3.1 Constrained Decoding
   3.2 Tree-to-sequence Model
Comcast’s Voice Interface for TV

22 million customers in 40 states of United States since 2015
http://www.xfinity.com/voice-remote

NLP IN THE CLOUD

1. Speech-to-Text
2. Intent Understanding
3. Action Resolution
4. Message Management

© Jinfeng Rao 2019
Watch ESPN
Show me free kids movies
When do the Phillies play?
Movies with Scarlett Johansson
Record Saturday Night Live
Challenges

• Short length of voice queries
  • Analysis over a week of 32M voice queries shows the average length is 2.04 words, much shorter than published stat. on smartphone & computers.

• Ambiguity
  • Query-level: a query “Chicago fires” can refer to either a television series or a soccer team.
  • Program-level: many TV programs share similar names: *The Princess Diaries, The Princess Diaries II: Royal Engagement, The Princess Knight*, etc…

• Many speech recognition errors
  • Little query/program lexical overlap because of SR errors, such as query “you” to program “Calliou”
Voice Session as Contexts

Find me free HBO shows.

1. Silicon Valley
2. West World
3. Blacklist

Find me Game of Throw

Do you mean the Fish Throw game or Game of Thrones?

© Jinfeng Rao 2019
**Session Analysis (~32M voice queries)**

**Session Length & Unsatisfactory Rate Distribution**

- **Observations**
  - More than 30% sessions have multiple queries, accounting for 57% of all queries.
  - The longer of a session, the higher of unsatisfactory rate
Problem Formulation

• Assumption
  • User will keep issuing queries until he find the intended program, i.e., [“hbo series”, “game of throw”, “game of throne”]

• Given a voice query session $S_i = [q_1, q_2, \ldots, q_n]$, we aim to predict the intended program at each query time, exploiting previous queries as contexts:

  \[
  \text{Data: } D = \{(s_i, p_i) \mid s_i = [q_{i_1}, \ldots, q_{i_{|s_i|}}], \ p_i \in \Phi\}_{i=1}^{D}
  \]

  \[
  \text{Model: } \hat{\theta} = \arg \max_{\theta} \prod_{i=1}^{D} \prod_{t=1}^{|s_i|} P(p_i \mid q_{i_1}, \ldots, q_{i_t}; \theta)
  \]

  where $S_i$ denotes a voice session, $P_i$ is the ground truth program,

  for each session  for each query
Bayesian Decomposition

Model: \( \hat{\theta} = \arg \max_{\theta} \prod_{i=1}^{D} \prod_{t=1}^{|s_i|} P(p_i|q_{i_1}, ..., q_{i_t}; \theta) \)

\[
P(p_i|q_{i_1}, ..., q_{i_t}) = P(p_i|c_{i_t}) \cdot P(c_{i_t}|v_{i_1}, ..., v_{i_t}) \cdot \prod_{j=1}^{t} P(v_{i,j}|q_{i_j})
\]

\[v_{i_t} \sim F(q_{i_t}; \theta_F), \quad c_{i_t} \sim G(v_{i_1}, ..., v_{i_t}; \theta_G), \quad p_i \sim H(c_{i_t}; \theta_H)
\]

prediction, contextual embedding, query embedding
Model: Hierarchical Recurrent Neural Networks (HRNN)

Program/Type Prediction

Query-level LSTM

Query Embedding Vector

Char-level LSTM

Char-level 1-Hot Vector

User’s Query Session

P('Game of Thrones') = 0.4

P('Game of Thrones') = 0.8

P('Game of Thrones') = 1.0

© Jinfeng Rao 2019
Model Variants

• Three ways of query representations
  • **character-level**: better for resolving speech recognition errors (i.e., “you” to “Caillou”)
  • **word-level**: better for semantic modeling
  • **combined**: combine the above two embedding ways

• Three model options
  • **basic**: remove the contextual modeling part
  • **full context**: full architecture, end-to-end training
  • **constrained context**: full architecture, pretrain the underlying layers, then fine-tune the top layers

• Total 3*3 model configurations
Data Collection

“Clickthrough”

“Watchthrough”
Experimental Setup

• Dataset

<table>
<thead>
<tr>
<th>Dataset</th>
<th>#sessions</th>
<th>#queries</th>
<th>Avg. session len</th>
<th>Avg. query len</th>
</tr>
</thead>
<tbody>
<tr>
<td>Train</td>
<td>126016</td>
<td>181058</td>
<td>1.44</td>
<td>2.34</td>
</tr>
<tr>
<td>Validation</td>
<td>28427</td>
<td>82828</td>
<td>2.91</td>
<td>2.30</td>
</tr>
<tr>
<td>Test</td>
<td>28173</td>
<td>82272</td>
<td>2.92</td>
<td>2.30</td>
</tr>
</tbody>
</table>

• Metrics
  • Precision at 1 (P@1), Precision at 5 (P@5), Mean Reciprocal Rank (MRR)

• Baselines
  • TF-IDF: match query with programs by character-level 3grams.
  • BM25
  • SVM-Rank with three sets of features: 1) BM25 score, 2) popularity prior of the program, and 3) the word embedding based similarity features.
  • Deep Structure Semantic Models (DSSM): 3gram based neural matching method
  • Basic model with three query representations.
## Main Results

<table>
<thead>
<tr>
<th>Model</th>
<th>Query</th>
<th>P@1</th>
<th>P@5</th>
<th>MRR</th>
</tr>
</thead>
<tbody>
<tr>
<td>TF-IDF</td>
<td>3-gram</td>
<td>0.518</td>
<td>0.593</td>
<td>0.543</td>
</tr>
<tr>
<td>BM25</td>
<td>3-gram</td>
<td>0.533</td>
<td>0.596</td>
<td>0.565</td>
</tr>
<tr>
<td>SVM-Rank</td>
<td>-</td>
<td>0.547</td>
<td>0.621</td>
<td>0.582</td>
</tr>
<tr>
<td>DSSM</td>
<td>-</td>
<td>0.568</td>
<td>0.617</td>
<td>0.584</td>
</tr>
<tr>
<td><strong>Our Approach</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Basic</td>
<td>char</td>
<td>0.605</td>
<td>0.647</td>
<td>0.690</td>
</tr>
<tr>
<td>Basic</td>
<td>word</td>
<td>0.609</td>
<td>0.644</td>
<td>0.677</td>
</tr>
<tr>
<td>Basic</td>
<td>comb</td>
<td>0.614</td>
<td>0.651</td>
<td>0.687</td>
</tr>
<tr>
<td>Full Context</td>
<td>char</td>
<td>0.482</td>
<td>0.532</td>
<td>0.580</td>
</tr>
<tr>
<td>Full Context</td>
<td>word</td>
<td>0.599</td>
<td>0.638</td>
<td>0.687</td>
</tr>
<tr>
<td>Full Context</td>
<td>comb</td>
<td>0.598</td>
<td>0.643</td>
<td>0.688</td>
</tr>
<tr>
<td>Cons-Context</td>
<td>char</td>
<td>0.639</td>
<td>0.684</td>
<td>0.731</td>
</tr>
<tr>
<td>Cons-Context</td>
<td>word</td>
<td>0.639</td>
<td>0.683</td>
<td>0.729</td>
</tr>
<tr>
<td><strong>Cons-Context</strong></td>
<td><strong>comb</strong></td>
<td><strong>0.643</strong></td>
<td><strong>0.687</strong></td>
<td><strong>0.734</strong></td>
</tr>
</tbody>
</table>

Two key factors:
Session Context Modeling
Query Combination
HRNN Context Analysis

- How does context signal help?
  - For each session length, we plot the average precision at different query position.
  - We take sessions with length of 6 as example

**Observations**
- Non-context methods (SVM-Rank, DSSM, Basic) stay flat at different query positions.
- Context methods consistently goes up with more queries as context.
# Case Study

<table>
<thead>
<tr>
<th>Session</th>
<th>Cacio -&gt; You -&gt; You -&gt; Calliou</th>
</tr>
</thead>
<tbody>
<tr>
<td>Program</td>
<td>Calliou</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Model</th>
<th>Query</th>
<th>Predictions</th>
</tr>
</thead>
<tbody>
<tr>
<td>BM25</td>
<td>3gram</td>
<td>Pacific Rim -&gt; Now You See Me -&gt; Now You See Me -&gt; 😊</td>
</tr>
<tr>
<td>SVM-Rank</td>
<td>-</td>
<td>Pacific Rim -&gt; Now You See Me -&gt; Now You See Me -&gt; 😊</td>
</tr>
<tr>
<td>DSSM</td>
<td>3gram</td>
<td>Pacific Rim -&gt; Young -&gt; Young -&gt; 😊</td>
</tr>
</tbody>
</table>

**Our Approach**

<table>
<thead>
<tr>
<th>Model</th>
<th>Query</th>
<th>Predictions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Basic</td>
<td>char</td>
<td>😊 (0.81) -&gt; 😊 (0.80) -&gt; 😊 (0.80) -&gt; 😊 (1.0)</td>
</tr>
<tr>
<td>Basic</td>
<td>word</td>
<td>Child Genius (0.03) -&gt; 😊 (0.57) -&gt; 😊 (0.57) -&gt; 😊 (1.0)</td>
</tr>
<tr>
<td>Basic</td>
<td>comb</td>
<td>Paw Patrol (0.17) -&gt; 😊 (0.83) -&gt; 😊 (0.83) -&gt; 😊 (1.0)</td>
</tr>
<tr>
<td>Cons-Context</td>
<td>char</td>
<td>😊 (0.96) -&gt; 😊 (0.99) -&gt; 😊 (0.99) -&gt; 😊 (1.0)</td>
</tr>
<tr>
<td>Cons-Context</td>
<td>word</td>
<td>Wallykazam (0.07) -&gt; 😊 (0.59) -&gt; 😊 (0.86) -&gt; 😊 (1.0)</td>
</tr>
<tr>
<td>Cons-Context</td>
<td>comb</td>
<td>Paw Patrol (0.17) -&gt; 😊 (0.93) -&gt; 😊 (1.0) -&gt; 😊 (1.0)</td>
</tr>
</tbody>
</table>

© Jinfeng Rao 2019
Product Deployment

• Run after a number of simpler NLP modules and processed 8% of total queries which the current system provided no response.

• Among 70M queries in a week, 5.7M queries were sent to our model, which provided response to 4.2M queries, increasing end-to-end coverage from 92% to 98%.

• Human annotation shows 2/3 response are relevant, while the left 1/3 are considered irrelevant.
Building a Comprehensive Voice Navigation Platform

• Drawbacks of the previous HRNN approach
  • Classification-based nature, unable to retrieve unseen TV programs
  • Can’t scale to hundreds of thousands of programs

• Towards a comprehensive voice navigation platform
  • In reality, users can say anything to their TVs (i.e., looking for movies, set reminders, checking the weather, etc.)
  • A comprehensive system that can response to *any general query*. 

© Jinfeng Rao 2019
What do user say to their TV?

1. Channel (30%): “HBO”
2. Movie/Series (27%): “Find me game of throne”
3. Event (9%): “Oscar show”
4. Browse (6%): “free action movies on HBO”
5. Record (1%): “record game of throne”
6. Command (1%): “turn on/off TV”
7. Hundreds of intents more…
Tag Distribution

Word-level Freq(%) of tags:
- Context
- Title
- Channel
- Cost
- Category
- Genre
- Team
- Person
- Format

Examples:
- "Find me game of throne"
  - Context
  - Title
- "Free action movies on HBO"
  - Cost
  - Genre
  - Category
  - Channel

© Jinfeng Rao 2019
Multi-Task Learning

- Given a voice session, perform three tasks:
  - **Program prediction**: predict the intended TV program if it exists
  - **Intent classification**: predict the query intent (movie/channel/browse/etc.)
  - **Tagging**: generate the word tag sequence for each query

- These three information together are necessary to understand an arbitrary query:
  - For simple queries, the three predictions should reinforce each other.
  - For ambiguous queries, combine the three evidences to interpret the query correctly.

“Find me game of throne”
  - intent = series
  - program = “game of throne”
  - tags – not important here

“Record game of throne”
  - intent = record
  - program = “game of throne”
  - tags - not important here

“Free action movies on HBO”
  - intent = browse
  - program = NA
  - tags = [cost, genre, category, context, channel]
Multi-Task Problem Formulation

• Given a voice query session \( S_i = [q_1, q_2, \ldots, q_n] \), we aim to predict the query intent, intended program and tag sequence at each query time, exploiting previous queries as contexts:

\[
\text{Data: } D = \{(s_i, p_i, A_i, T_i) \mid s_i = [q_{i_1}, \ldots, q_{i_n}], \ p_i \in \Phi,
\]

\[
A_i = [a_{i_1}, \ldots, a_{i_n}], \ T_i = [\tau_{i_1}, \ldots, \tau_{i_n}]\}_{1}^{\left| D \right|}
\]

\[
\text{Model: } \hat{\theta} = \arg \max_{\theta} \prod_{i=1}^{\left| D \right|} \prod_{t=1}^{n} P(p_i, A_i, T_i | q_{i_1}, \ldots, q_{i_t}; \theta)
\]

where \( S_i \) denotes a voice session, \( P_i \) is the intended program (one per session), \( A_i \) is the query intent sequence (one per query), \( T_i \) is the tag sequences (one tag per word)
Bayesian Decomposition

Model: \( \hat{\theta} = \arg \max_{\theta} \prod_{i=1}^{|D|} \prod_{t=1}^{n} P(p_i, A_i, \mathcal{T}_i | q_{i_1}, \ldots, q_{i_t}; \theta) \)

\( = \arg \max_{\theta} \prod_{i=1}^{|D|} \prod_{t=1}^{n} P(p_i | c_{i_t}) \cdot P(a_{i_t} | c_{i_t}) \cdot P(\tau_{i_t} | q_{i_t}) \cdot P(c_{i_t} | v_{i_1}, \ldots, v_{i_t}) \cdot \prod_{j=1}^{t} P(v_{i_j} | q_{i_j}) \)

- program prediction
- intent classification
- tagging
- contextual embedding
- query embedding

© Jinfeng Rao 2019
Multi-Task Learning Framework

- Query embedding component
  - Lookup layer + BiLSTM

- Contextual embedding comp
  - another LSTM

- Task-specific components
  - Program prediction
  - Intent classification
  - Tagging

© Jinfeng Rao 2019
Program Prediction

- A ranking formulation
  \[ P(p_i | c_{i_t}) \rightarrow P(\text{rel} | c_{i_t}, p^+) > P(\text{rel} | c_{i_t}, p^-), \forall p^- \in \Phi \]

- Model query and program jointly
  - Search-based program representation
  - Title-based
  - Combination-based

- Contrastive list-wise loss to maximize the pos relevance given all negatives
  \[
P(p^+ | q_{i_1}, ..., q_{i_t}) = \frac{\exp(P(\text{rel} | p^+, q_{i_1}, ..., q_{i_t}))}{\sum_{p^- \in C} \exp(P(\text{rel} | p^-, q_{i_1}, ..., q_{i_t}))}
\]
  \[
P(\text{rel} | c_{i_t}, p) = \text{cosine}(c_{i_t}, p)
\]
Intent Classification & Tagging

• Classification modeling since the intent vocab is small and stable
  • Modeled using a fully connected layer + cross-entropy loss

• Tagging is a sequential labeling task
  • Modeled using a conditional random field (CRF)
  • MLE for training and Viterbi algorithm for decoding
Multi-Task Optimization

• Stage 1: weighted sum of the losses of three tasks and optimize simultaneously

\[ L = w_p \cdot L_p + w_i \cdot L_i + w_t \cdot L_t \]

• Stage 2: fix the underlying shared components (the query embedding + contextual layer), fine-tune the task-specific layer with their own loss.
## Experimental Setup

<table>
<thead>
<tr>
<th>Dataset</th>
<th>#sessions</th>
<th>#queries</th>
<th>Avg. session len</th>
<th>Avg. query len</th>
</tr>
</thead>
<tbody>
<tr>
<td>Train</td>
<td>0.87M</td>
<td>1.2M</td>
<td>1.36</td>
<td>2.34</td>
</tr>
<tr>
<td>Validation</td>
<td>0.62M</td>
<td>0.82M</td>
<td>1.32</td>
<td>2.30</td>
</tr>
<tr>
<td>Test</td>
<td>0.62M</td>
<td>0.82M</td>
<td>1.33</td>
<td>2.30</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>#programs</th>
<th>#channels</th>
<th>#intents</th>
<th>#tags</th>
<th>single-query vs. multi-query session</th>
<th>Unseen sessions (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>26247</td>
<td>244</td>
<td>109</td>
<td>11</td>
<td>80:20</td>
<td>10%</td>
</tr>
</tbody>
</table>

- We reused the baselines and evaluation metrics in the previous HRNN model.
# Main Results

<table>
<thead>
<tr>
<th>Method</th>
<th>Program P@1</th>
<th>Program P@5</th>
<th>Intent Acc. (%)</th>
<th>Tagging Acc. (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>TF-IDF</td>
<td>0.663</td>
<td>0.745</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>BM25</td>
<td>0.674</td>
<td>0.750</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>SVM-Rank</td>
<td>0.682</td>
<td>0.758</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>DSSM</td>
<td>0.703</td>
<td>0.765</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Stanford CRF Tagger</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>0.821</td>
</tr>
</tbody>
</table>

Our Previous Work in CIKM’17

<table>
<thead>
<tr>
<th>Method</th>
<th>Program P@1</th>
<th>Program P@5</th>
<th>Intent Acc. (%)</th>
<th>Tagging Acc. (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>HRNN w/ LSTM</td>
<td>0.724</td>
<td>0.783</td>
<td>0.915</td>
<td>0.884</td>
</tr>
<tr>
<td>HRNN w/ BiLSTM</td>
<td>0.725</td>
<td>0.786</td>
<td>0.916</td>
<td>0.939</td>
</tr>
</tbody>
</table>

Our Approach

<table>
<thead>
<tr>
<th>Method</th>
<th>Program P@1</th>
<th>Program P@5</th>
<th>Intent Acc. (%)</th>
<th>Tagging Acc. (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single-Task Learning</td>
<td>0.738</td>
<td>0.801</td>
<td>0.917</td>
<td>0.944</td>
</tr>
<tr>
<td>Multi-Task Learning</td>
<td>0.757</td>
<td>0.824</td>
<td>0.925</td>
<td>0.945</td>
</tr>
</tbody>
</table>
## Error Analysis: Program Prediction

<table>
<thead>
<tr>
<th>Method</th>
<th>Program P@1</th>
<th>Program P@5</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVM-Rank</td>
<td>0.682</td>
<td>0.758</td>
</tr>
<tr>
<td>DSSM</td>
<td>0.703</td>
<td>0.765</td>
</tr>
<tr>
<td>HRNN</td>
<td>0.725</td>
<td>0.786</td>
</tr>
<tr>
<td><strong>Single-Task Learning</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Search-based</td>
<td>0.715</td>
<td>0.770</td>
</tr>
<tr>
<td>Title-based</td>
<td>0.720</td>
<td>0.796</td>
</tr>
<tr>
<td>Combination-based</td>
<td>0.738</td>
<td>0.801</td>
</tr>
<tr>
<td><strong>Multi-Task Learning</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Search-based</td>
<td>0.731</td>
<td>0.790</td>
</tr>
<tr>
<td>Title-based</td>
<td>0.728</td>
<td>0.808</td>
</tr>
<tr>
<td>Combination-based</td>
<td>0.757</td>
<td>0.824</td>
</tr>
</tbody>
</table>

- **Observations**
  - Ranking-based approaches suffered
    - Lexical mismatch of query/program
    - Speech recognition errors
  - Combination-based is effective
  - Multi-task learning improved the performance by a lot
Error Analysis: Intent Classification and Tagging

Intent Classification

<table>
<thead>
<tr>
<th>Intent</th>
<th>Channel</th>
<th>Movie</th>
<th>Series</th>
<th>Event</th>
<th>Browse</th>
</tr>
</thead>
<tbody>
<tr>
<td>Channel</td>
<td>97.9%</td>
<td>0.4%</td>
<td>0.5%</td>
<td>0</td>
<td>0.2%</td>
</tr>
<tr>
<td>Movie</td>
<td>0.4%</td>
<td>89.7%</td>
<td>2.4%</td>
<td>0.1%</td>
<td>3.3%</td>
</tr>
<tr>
<td>Series</td>
<td>0.2%</td>
<td>0.8%</td>
<td>96.2%</td>
<td>0</td>
<td>1.3%</td>
</tr>
<tr>
<td>Event</td>
<td>0.4%</td>
<td>3.7%</td>
<td>1.6%</td>
<td>87.3%</td>
<td>0</td>
</tr>
<tr>
<td>Browse</td>
<td>0.1%</td>
<td>1.9%</td>
<td>1.8%</td>
<td>0</td>
<td>94.1%</td>
</tr>
</tbody>
</table>

Table 4: Confusion matrix for the top 5 intent types.

• Findings:
  • Event intent is the most ambiguous one because of its blurred definition.
  • For tagging, neural models win the Stanford CRF tagger in popular tags, while lose in the infrequent tags.

Accuracy distributions for the top 5 tags.
20 million voice-enabled remotes delivered to customers across the US

9 billion+ voice commands processed in 2019

MISC: Won the 69th Emmy Award (2017) for the technical contribution in advancing television technologies.
Outline

1. Textual Semantic Modeling
   1.1 Multi-perspective relevance matching
   1.2 Bridging the gap of relevance and semantic matching
   1.3 Quaternion Transformer

2. Temporal Context Modeling
   2.1 Query trend vs. term trend
   2.2 Session modeling with multi-task learning

3. Structural Modeling for NLG
   3.1 Constrained Decoding with Compositional MR
   3.2 Tree-to-sequence Model
Related Papers in Part 3

1. J. Rao, K. Upasani, A. Balakrishnan, M. White, R. Subba, Tree-to-Sequence Model for Neural NLG in Task-Oriented Dialog, INLG 2019

Background: E2E NLG Dataset

- ~50K (meaning representation (MR), sentence) pairs for the restaurants domain

**MR:** name[JJs’ Pub] rating[5 out of 5] familyFriendly[no] eatType[restaurant] near[Crowne Plaza Hotel]

**Sentence:** JJ’s pub is not a family friendly restaurant. It has a high customer rating of 5 out of 5. You can find it near the Crowne Plaza Hotel.
E2E Dataset and Shortcomings

• Provided crowdworkers with MR and asked them to write responses

• Lots of diversity in dataset, e.g. argument grouping, contrast, rich language
  • [But hard to control these aspects]

• However, models trained on the data lack this diversity
  • e.g. contrast only occurs 0.4% of the time in model generations

Possible reason: Same MR → different discourse structures
Example of E2E Dataset


Reference 1: JJ’s pub is not a family friendly restaurant. It has a high customer rating of 5 out of 5. You can find it near the Crowne Plaza Hotel.

Reference 2: JJ’s Pub is not family friendly, but it has a high customer rating of 5 out of 5. It is a restaurant near the Crowne Plaza Hotel.
Reed et al. (2018)

• Proposed adding tokens to indicate target discourse structure (contrast/no contrast, # of sentences)

• Greatly improved accuracy of contrast expressions and # sentences

• But: no way to control which slots gets contrasted!

How can we control the expression of discourse relations?
Why control discourse?

- Prior work has shown increases in **perceived naturalness** when user models or preferences influence discourse.

---

What can you tell me about JJ’s Pub? I’d like to eat English food.

JJ’s pub is highly rated, but it serves English food.

5/5 rating and English food are usually contrasted

Discourse-unaware system

Lemon et al. 2014; Carenin and Moore, 2006; Walker et al. 2007, White et al. 2010; Demberg et al. 2011
Why control discourse?

- Prior work has shown increases in **perceived naturalness** when user models or preferences influence discourse.

Lemon et al. 2014; Carenin and Moore, 2006; Walker et al. 2007, White et al. 2010; Demberg et al. 2011
Proposed Compositional Meaning Representations

- **Arguments** e.g. date_time, family_friendly hah
  - Entities or slots to be mentioned
  - Can be nested (e.g., date_time contains week_day)

- **Dialog Acts**, e.g., INFROM, RECOMMEND
  - Contain arguments
  - An atomic unit to be expressed in an utterance

- **Discourse relations**, e.g., JUSTIFY, CONTRAST
  - Relationship between dialog act and discourse relations

© Jinfeng Rao 2019
Put it all together

• Tree-structured representation

• Allows arbitrary nesting of relations and acts

```
[JOIN
 [CONTRAST
  [INFORM [name JJs' Pub] [rating 5 out of 5] ]
  [INFORM [familyFriendly no ] ]
 ]
[INFORM [eatType restaurant] [near Crowne Plaza Hotel] ]
```
Data Creation

Rule-based

User query crowdsourcing → Compositional MR generation → Response generation and annotation → Response quality evaluation → 33K examples for *weather* domain
Data Creation

Rule-based

User query crowdsourcing → Compositional MR generation → Response generation and annotation → Response quality evaluation → 33K examples for weather domain

**User context**

Date: October 5
Location: Austin

**Query**

How heavy is the rain expected to be?

INFORM[
  condition_not[rain]
  date_time[ month[October] day[5] ]
  location[ city[Austin] ]
]

**MR**

INFORM[
  cloud_coverage[partly cloudy]
  temp_high[50] temp_low[37]
  date_time[ month[October] day[5] ]
  location[ city[Austin] ]
]

**Annotator**

No rain is expected today in Austin. It’ll be partly cloudy with a high of 50 and a low of 37.

[INFORM No [CONDITION_NOT rain ] is expected [DATE_TIME [COLLOQUIAL today ] in [LOCATION [CITY Austin ]]]. [INFORM It’ll be [CLOUD_COVERAGE partly cloudy ] with a high of [TEMP_HIGH 50 ] and a low of [TEMP_LOW 37 ]].]
Modified E2E Dataset

• Used the Berkeley neural parser to automatically infer CONTRAST and JOIN relations in the E2E challenge dataset

• Used Slug2Slug token tagger and HarvardNLP latent segmentation model to annotate responses
Model Overview

Standard Seq2Seq model with attention

**Input:** Linearized tree structure

```
[CONTRAST [INFORM [condition rain ] [date_time [week_day Saturday ] ] ] [INFORM [cloud_coverage sunny ] [date_time [week_day Sunday ] ] ]]
```

**Output:** Response with tree structure preserved

```
[CONTRAST [INFORM There’ll be [condition rain ] on [date_time [week_day Saturday ] ] ], but [INFORM [date_time [week_day Sunday ] ] will be [cloud_coverage sunny ] ]].
```
Semantic Correspondence

Leverage correspondence between input and output structures

[CONTRAST [INFORM [condition rain] [date_time [week_day Saturday]]] [INFORM [cloud_coverage sunny] [date_time [week_day Sunday]]]]

[JOIN [INFORM There’ll be [condition rain] on [date_time [week_day Saturday]], and [INFORM [date_time [week_day Sunday]] will be [cloud_coverage sunny]]]].
Semantic Correspondence

Leverage correspondence between input and output structures

[CONTRAST [INFORM [condition rain ] [date_time [week_day Saturday ] ] ] [INFORM [cloud_coverage sunny ] [date_time [week_day Sunday ] ] ] ] ]

Tree accuracy metric for semantic correctness

[JOIN [INFORM There’ll be [condition rain ] on [date_time [week_day Saturday ] ] ], and [INFORM [date_time [week_day Sunday ] ] will be [cloud_coverage sunny ] ] ].
Constrained Decoding

Idea: During beam search, invalidate tokens that would result in responses that don’t match the input structure (based on the nonterminals in the output)

```
[INFORM [condition_not rain] [date_time [colloquial today ] ] ]
```
Constrained Decoding (Continued)

• Invalidate tokens that would result in responses that don’t match the input tree structure
  • Don’t allow opening non-terminals (e.g., [CONTRAST) that aren’t allowed in that part of the subtree
  • Don’t allow closing non-terminals (e.g., ]) if subtree isn’t complete
    • Account for ellipsis/aggregation (if a value has already been expressed previously)
Experiments Setup

Baselines:

• FLAT
  • Input and output contain a flat list of arguments
• TOKEN
  • Inspired by Reed et al. 2018, input is similar to FLAT but add tokens indicating # of Contrast and Join relations
• TREE
  • Input and output are linearized tree representations
• CONSTR
  • Tree with constrained decoding
Experiments Setup

Metrics:

• Automatic
  • BLEU score
  • Tree accuracy (binary)
• Human eval
  • Grammaticality
  • Semantic Correctness (argument missing, repetition, grouping, etc.)
  • Disc. Correctness (semantic correctness measured on challenge subset)
Main Results

<table>
<thead>
<tr>
<th>Model Metric</th>
<th>BLEU</th>
<th>TreeAcc</th>
<th>E2E Gram</th>
<th>Corr</th>
<th>Disc</th>
<th>BLEU</th>
<th>TreeAcc</th>
<th>Weather Gram</th>
<th>Corr</th>
<th>Disc</th>
</tr>
</thead>
<tbody>
<tr>
<td>S2S-FLAT</td>
<td>0.6360</td>
<td>-</td>
<td>94.03</td>
<td>63.85</td>
<td>30.87</td>
<td>0.7455</td>
<td>-</td>
<td>98.77</td>
<td>77.09</td>
<td>79.04</td>
</tr>
<tr>
<td>S2S-TOKEN</td>
<td>0.7441†</td>
<td>-</td>
<td>92.29</td>
<td>69.02†</td>
<td>42.29†</td>
<td>0.7493*</td>
<td>-</td>
<td>96.7</td>
<td>81.56†</td>
<td>83.93†</td>
</tr>
<tr>
<td>S2S-TREE</td>
<td>0.7458†</td>
<td>94.86</td>
<td>93.59</td>
<td>83.85†</td>
<td>54.35†</td>
<td>0.7612*</td>
<td>92.5</td>
<td>95.26</td>
<td>87.61†</td>
<td>85.97†</td>
</tr>
<tr>
<td>S2S-CONSTR</td>
<td>0.7469†</td>
<td>99.25</td>
<td>94.33</td>
<td>85.89†</td>
<td>66.09†</td>
<td>0.7660*</td>
<td>96.92</td>
<td>95.30</td>
<td>91.82†</td>
<td>93.44†</td>
</tr>
</tbody>
</table>

- Tree-based representations result in responses are more consistent with the desired structure
- Constrained decoding improves semantic correctness further
Observations

• Constrained decoding can still generate incorrect responses
  • Model generates surface forms without generating non-terminals
  • [restaurant <name> ] is **not family friendly** and serves [cuisine Indian ] food
• Occasional (< 1%) stuttering due to constrained decoding
• Unnatural phrasings
  • Yes, you should wear [ATTIRE an umbrella ].
Generalization Ability

- Each dataset contains
  - Flat (simple)
  - Compositional (hard)
Outline

1. Textual Semantic Modeling
   1.1 Multi-perspective relevance matching
   1.2 Bridging the gap of relevance and semantic matching
   1.3 Quaternion Transformer

2. Temporal Context Modeling
   2.1 Query trend vs. term trend
   2.2 Session modeling with multi-task learning

3. Structural Modeling for NLG
   3.1 Constrained Decoding with Compositional MR
   3.2 Tree-to-sequence Model
Research Question

Would combining a better learning and decoding algorithm that leverage tree structures lead to better effectiveness?

Tree structured input:

```
INFORM [ condition[sunny], date.time.range[ colloquial[this weekend ] ] ]
CONTRAST [ ]
  INFORM [ avg.high[60s] date.time[ colloquial this weekend ] ]
INFORM [ chance[likely], wind.summary[heavy], date.time[ week.day[Saturday] colloquial[morning] ] ]
```
Tree-to-Sequence Model

- Tree-based Encoder
  - N-ary TreeLSTM encoder [1]

\[ h_k^D = f_{\text{tree}} (\{ h_k^{c_1}, \ldots, h_k^{c_N} \}) \]

- Trees are hard to parallelize
  - Develop an iterative bottom-up traversal algorithm to support batch forward/backward
  - 5-10X speedup in training and inference
Tree-to-Sequence Model

• Structure-aware Decoder

  • Initialize decoder state as root hidden state
    \[ s_1 = h_{\text{root}} \]

  • Incorporate tree structures into decoding

\[
\alpha_i(k) = \frac{\exp(s_i \cdot h_k)}{\sum_{k=1}^{K} \exp(s_i \cdot h_k)}
\]

\[
d_j = \sum_k \alpha_i(k) h_k;
\]

\[
\hat{s}_j = \tanh(W_d \cdot [s_j; d_j; s_{\hat{j}-1}] + b_d)
\]

\[
P(y_i | y_{<i}, x) = (W_s \cdot \hat{s}_j + b_s)
\]
Main Results

<table>
<thead>
<tr>
<th>Model Metric</th>
<th>BLEU</th>
<th>E2E NoDISCACC</th>
<th>DISCACC</th>
<th>BLEU</th>
<th>Weather NoDISCACC</th>
<th>DISCACC</th>
</tr>
</thead>
<tbody>
<tr>
<td>S2S</td>
<td>74.58</td>
<td>99.68</td>
<td>95.28</td>
<td>76.75</td>
<td>96.62</td>
<td>83.3</td>
</tr>
<tr>
<td>S2S-CONSTR</td>
<td>74.69</td>
<td><strong>99.89</strong></td>
<td>97.78</td>
<td>77.45</td>
<td>98.52</td>
<td>91.61</td>
</tr>
</tbody>
</table>

Our Approaches

<table>
<thead>
<tr>
<th></th>
<th>BLEU</th>
<th>E2E NoDISCACC</th>
<th>DISCACC</th>
<th>BLEU</th>
<th>Weather NoDISCACC</th>
<th>DISCACC</th>
</tr>
</thead>
<tbody>
<tr>
<td>T2S</td>
<td>74.75</td>
<td><strong>99.89</strong></td>
<td>96.96</td>
<td>77.86</td>
<td>97.1</td>
<td>88.8</td>
</tr>
<tr>
<td>T2S-CONSTR</td>
<td>74.63</td>
<td>99.84</td>
<td><strong>98.60</strong></td>
<td>77.82</td>
<td><strong>99.11</strong></td>
<td>94.13</td>
</tr>
</tbody>
</table>

- T2S (Tree2Seq) is better on tree accuracy, especially on the DISC subset (more challenging subset)
- Combine T2S and constrained decoding further improves effectiveness
Model Effectiveness Study

Tree Acc by # DialogActs

Weather Tree Accuracy Distribution

Tree Acc by % data

Weather Data Efficiency Study

© Jinfeng Rao 2019
Summary

• I present different ways to modeling relevance and semantics for multiple IR and NLP tasks.

• I present families of modeling techniques to model different granularities of time information and demonstrated their effectiveness on various applications.

• I present novel learning and decoding algorithms to leverage tree structures for neural NLG
Thanks!

Q & A
Reference

2. J. Rao, H. He, J. Lin, Integrating Lexical and Temporal Signals in Neural Ranking Models for Social Media Search, SIGIR NeuIR 2017
5. J. Rao, V. Yang, Y. Zhang, F. Ture, J. Lin, Multi-Perspective Relevance Matching with Hierarchical ConvNets for Social Media Search, arXiv 1805.08519
7. Multi-Task Learning with Neural Networks for Voice Query Understanding on an Entertainment Platform, KDD 2018
8. Xiaoyan Li and W. Bruce Cro, Time-Based Language Models, CIKM 2003, 469–475.

© Jinfeng Rao 2019
Reference

16. Hua He, Kevin Gimpel, and Jimmy Lin, 2014, Multi-Perspective Sentence Similarity Modeling with Convolutional Neural Networks, in EMNLP
17. Jiafeng Guo, Yixing Fan, Qingyao Ai, W Bruce Croft, 2016, A Relevance Matching Model for Ad-hoc Retrieval, in CIKM
Compression of Query Trends

- Query trends = number of terms * number of time intervals (i.e., every five mins)
- Large but sparse, amenable for compression

Compression methods:
1. Variable-byte Encoding (VB) [20]
2. Simple16 [21]
3. PForDelta (P4D) [22]
4. Discrete Wavelet Transform
5. Huffman Encoding (ours [3]):
   5.1 Partition a list of integers to continuous blocks, with each block having 8 consecutive integers.
   5.2 Construct a Huffman tree based on frequencies of blocks.
   5.3 Concatenate the binary Huffman codes of blocks, then convert back to compact integer list.
Evaluation of Compression

Compression on Tweets2013 corpus (243M tweets)
Evaluated on compression size and decoding time.

<table>
<thead>
<tr>
<th>Tweets2013</th>
<th>Unigrams</th>
<th>Bigrams</th>
</tr>
</thead>
<tbody>
<tr>
<td>Method</td>
<td>size (GB)</td>
<td>percentage</td>
</tr>
<tr>
<td>Raw</td>
<td>52.5</td>
<td></td>
</tr>
<tr>
<td>VB</td>
<td>13.1</td>
<td>+446 %</td>
</tr>
<tr>
<td>Simple16</td>
<td>2.2</td>
<td>-8.33 %</td>
</tr>
<tr>
<td>P4D</td>
<td>2.4</td>
<td>-</td>
</tr>
<tr>
<td>Wavelet+VB</td>
<td>14.8</td>
<td>+517 %</td>
</tr>
<tr>
<td>Wavelet+P4D</td>
<td>3.8</td>
<td>+58.3 %</td>
</tr>
<tr>
<td>Huffman</td>
<td>0.71</td>
<td>-70.4 %</td>
</tr>
<tr>
<td>Huffman+VB</td>
<td>0.48</td>
<td>-80.0 %</td>
</tr>
<tr>
<td>Optimal</td>
<td>0.33</td>
<td>-86.2 %</td>
</tr>
</tbody>
</table>

Observation:
1. Huffman based methods achieve state-of-the-art compression sizes, while still maintaining acceptable decoding speed.
2. The performance of Huffman-based method are close to the optimal.
HRNN Data Generation

• Collected ~32M raw voice queries during Feb. 22-28th 2016 from 2.5M unique viewers in the Comcast’s Xfinity X1 platform.

• Selected 45 seconds to sessionize queries, yielding ~20M sessions.

• The ground truth labels of sessions are collected by viewers’ watching behaviors after their queries, yielding 13M session-program pairs.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>#sessions</th>
<th>#queries</th>
<th>Avg. session len</th>
<th>Avg. query len</th>
</tr>
</thead>
<tbody>
<tr>
<td>Train</td>
<td>126016</td>
<td>181058</td>
<td>1.44</td>
<td>2.34</td>
</tr>
<tr>
<td>Validation</td>
<td>28427</td>
<td>82828</td>
<td>2.91</td>
<td>2.30</td>
</tr>
<tr>
<td>Test</td>
<td>28173</td>
<td>82272</td>
<td>2.92</td>
<td>2.30</td>
</tr>
</tbody>
</table>

© Jinfeng Rao 2019
HRNN Training Details

• GloVe word embeddings
  • 1.8K/20.4K words were not found in the vocabulary
  • Unseen word embeddings were randomly initialized from [-0.05, 0.05]

• RMS-PROP algorithm for parameter updating
  • learning rate was set to 0.001 initially, then decreased a factor of 3 if validation set loss
doesn’t go down for three epoches.

• Categorical loss function was adopted

• At test time, we selected the model that obtained the highest P@1 accuracy at
validation set for evaluation.
Conclusion

• As the first time, we present a comprehensive voice search system that allows users talk to their TVs.

• We decompose the system into jointly learning three tasks: program prediction, intent classification and tagging.

• Experiments on large real datasets show multi-task learning can improve the performance than learning individually.

• Our work was also awarded by the 69th Emmy Award (2017) for the technical contribution in advancing television technologies.