MeLL: Large-scale Extensible User Intent Classification for Dialogue Systems with Meta Lifelong Learning

Chengyu Wang\textsuperscript{1*}, Haojie Pan\textsuperscript{1*}, Yuan Liu\textsuperscript{1}, Kehan Chen\textsuperscript{1}, Minghui Qiu\textsuperscript{1}, Wei Zhou\textsuperscript{1}, Jun Huang\textsuperscript{1}, Haiqing Chen\textsuperscript{1}, Wei Lin\textsuperscript{1}, Deng Cai\textsuperscript{2}

\textsuperscript{1} Alibaba Group \textsuperscript{2} State Key Lab of CAD & CG, Zhejiang University
Introduction (1)

✓ User Intent Classification: **Text-to-label Classifiers**
  - Understanding users’ intents based on the input queries issued by users
  - Understanding users’ responses to actions previously taken by the systems

✓ Extensible User Intent Classification
  - The task number is continuously growing through time

✓ Challenges
  - **Parameter explosion**
  - **Catastrophic forgetting**
Solution: the Meta Lifelong Learning (MeLL) framework

Components

- Text Encoder
- Global Memory
- Local Memories
- Task-specific Layers
Introduction (3)

✔ Functionalities of Different Components

• Text Encoder: learning the semantics of input texts (slowly updated)
• Global Memory: storing the class semantics across tasks (fast updated)
• Local Memories: storing the task-specific class semantics (frozen once assigned)
• Task-specific Layers: generating task-specific outputs
Related Work

- User Intent Classification
- Lifelong Learning
  - Solving an unlimited sequence of tasks with the help of previously learned tasks
- Meta-learning
  - Training meta-learners that can adapt to a variety of tasks with little training data available
- Pre-trained Language Models

MeLL: leveraging ideas of both lifelong learning and meta-learning for user intent classification based on pre-trained language models
MeLL: Basic Model Structure

- **Data/Tensor Flows (Forward Pass)**
  - Task 1
  - Task 2
  - Tasks 1 & 2

- **Task-specific Output Layers**

- **Attentive Feature Fusion**

- **Global Memory Network** (Fast-updated, w. LRU Replacement Policy)

- **Text Encoder (Slowly-updated)**

- **Local Memory Network (Frozen)**

- **Select & Freeze**

- **Task Meta-Info.**

- **Input Text**

- **a) MeLL (Training Time)**

- **b) MeLL (Inference Time)**
MeLL (Training Time)

Algorithm 1 MeLL Training Procedure

1: // Initial Learning Stage
2: Initialize global memory \( G \) based on \( \mathcal{D}_1, \mathcal{D}_2, \cdots, \mathcal{D}_N \).
3: while not converge do
4:     Sample a task \( \mathcal{T}_n \) from \( \mathcal{T}_1, \mathcal{T}_2, \cdots, \mathcal{T}_N \).
5:     Read a batch \( \{(x_{n,i}, y_{n,i})\} \) from \( \mathcal{D}_n \).
6:     Run through BERT to obtain representations \( \{Q(x_{n,i})\} \).
7:     Read global memory \( G \) with the task meta-info, \( \mathcal{Y}_n \) and text representations \( \{Q(x_{n,i})\} \) to generate features \( \{\text{Att}(x_{n,i})\} \) and pass them to the output layer \( f_n \).
8:     Update parameters of \( f_n, G \) and the text encoder by back propagation.
9: end while
10: Create local memories \( L_1, L_2, \cdots, L_N \) for \( \mathcal{T}_1, \mathcal{T}_2, \cdots, \mathcal{T}_N \), with all parameters frozen.
11: // Lifelong Learning Stage (Assume task \( \mathcal{T}_j \) arrives, \( j > N \))
12: Update global memory \( G \) based on \( \mathcal{D}_j \) w. LRU replacement.
13: Train the model with a new task-specific output layer \( f_j \) and a smaller learning rate on BERT. Parameters of \( f_n, G \) and BERT are updated.
14: Create local memory \( L_j \) for \( \mathcal{T}_j \) with all parameters frozen.
MeLL (Inference Time)

**Algorithm 2 MeLL Inference Procedure**

1. Read a batch \(\{(x_{n,i})\}\) from an unlabeled dataset of task \(T_n\).
2. Run through BERT to obtain representations \(\{Q(x_{n,i})\}\).
3. Read local memory \(L_n\) with the task meta-info. \(Y_n\) and text representations \(\{Q(x_{n,i})\}\) to generate features \(\{Att(x_{n,i})\}\) and pass them to the task-specific output layer \(f_n\).
4. Make predictions \(\{\hat{y}_{n,i}\}\) based on \(f_n(Att(x_{n,i}))\).
Global and Local Memory Networks

Global Memory Network

- Each “slot” stores the “centroid” representation for each class.

Initial Stage
\[
G_N^{(m)} = \frac{1}{|\mathcal{T}^{(m)}|} \sum_{\mathcal{T}_n \in \mathcal{T}^{(m)}} \frac{1}{|\mathcal{D}_n^{(m)}|} \sum_{(x_{n,i}, y_{n,i}) \in \mathcal{D}_n^{(m)}} Q(x_{n,i})
\]

Update Rule
\[
G_j^{(m)} = (1 - \gamma)G_{j-1}^{(m)} + \frac{\gamma}{|\mathcal{D}_j^{(m)}|} \sum_{(x_{n,i}, y_{n,i}) \in \mathcal{D}_j^{(m)}} Q(x_{n,i})
\]

- Replacement policy for “slots”: Least Recently Used (LRU)

Hyper-parameter Settings: \( N = 2, K = 4 \)
Feature Fusion and Model Output

✓ Feature Fusion

• Attentive score
  \[ \alpha^{(m)}(x_{n,i}) = \text{softmax}(Q(x_{n,i})^T \cdot G_{n}^{(m)}) \]

• Attentive features
  \[ \text{Att}(x_{n,i}) = Q(x_{n,i}) + \sum_{y^{(m)} \in \mathcal{Y}_n} \alpha^{(m)}(x_{n,i}) \cdot G_{n}^{(m)} \]

✓ Model Output

• Each task has its own task-specific output layer.

Results from BERT encoder

Results from global memory
Experiments (1)

✓ Datasets
  • TaskDialog-EUIC: built from three public query intent classification datasets
  • Hotline-EUIC: a real-world e-commerce dataset for response intent classification in hotline agents

✓ Experimental Settings
  • bert-base-en (uncased) for TaskDialog-EUIC
  • roberta-tiny-chinese for Hotline-EUIC

<table>
<thead>
<tr>
<th></th>
<th>TaskDialog-EUIC</th>
<th>Hotline-EUIC</th>
</tr>
</thead>
<tbody>
<tr>
<td>#Train.</td>
<td>12,845</td>
<td>90,594</td>
</tr>
<tr>
<td>#Dev.</td>
<td>2,569</td>
<td>10,114</td>
</tr>
<tr>
<td>#Test</td>
<td>2,569</td>
<td>11,803</td>
</tr>
<tr>
<td>#Tasks</td>
<td>90</td>
<td>90</td>
</tr>
<tr>
<td>#Base tasks</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>#Distinct labels</td>
<td>26</td>
<td>71</td>
</tr>
</tbody>
</table>
# Experiments (2)

- Examples of Hotline-EUIC

<table>
<thead>
<tr>
<th>Domain</th>
<th>Task Description</th>
<th>User Response Intents</th>
</tr>
</thead>
<tbody>
<tr>
<td>Map</td>
<td>Check whether the shop name is correct</td>
<td>{Yes, No, Other}</td>
</tr>
<tr>
<td></td>
<td>Check whether the shop is still open</td>
<td>{Open, Close, Not sure}</td>
</tr>
<tr>
<td>Health</td>
<td>Ask about the medication history</td>
<td>{1 Year, 1-3 Years, &gt;3 Years}</td>
</tr>
<tr>
<td></td>
<td>Ask about the fasting plasma glucose</td>
<td>{Normal, Pre-diabetes, Diabetes}</td>
</tr>
<tr>
<td>Food takeout</td>
<td>Check if the customer is available to pick up the takeout</td>
<td>{Available, Not available, Deliver as soon as possible}</td>
</tr>
<tr>
<td></td>
<td>Satisfaction survey</td>
<td>{Satisfied, Slow delivery, Food spilled, Not received}</td>
</tr>
<tr>
<td>Express delivery</td>
<td>Check if the customer is available to pick up the delivery</td>
<td>{Available, Not available, Collect the parcels by others}</td>
</tr>
<tr>
<td></td>
<td>Satisfaction survey</td>
<td>{Satisfied, Slow delivery, Package damaged, Not received}</td>
</tr>
</tbody>
</table>
Experiments (3)

✓ Overall Model Performance

<table>
<thead>
<tr>
<th>Task</th>
<th>All tasks</th>
<th>New tasks</th>
<th>All tasks</th>
<th>New tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Accuracy</td>
<td>F1</td>
<td>Accuracy</td>
<td>F1</td>
</tr>
<tr>
<td>Results</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MTL (Upper-bound)*</td>
<td>0.9597</td>
<td>0.9590</td>
<td>0.9568</td>
<td>0.9562</td>
</tr>
<tr>
<td>Single*</td>
<td>0.9006</td>
<td>0.8974</td>
<td>0.9005</td>
<td>0.8969</td>
</tr>
<tr>
<td>Lifelong-freeze</td>
<td>0.9214</td>
<td>0.9194</td>
<td>0.9015</td>
<td>0.8988</td>
</tr>
<tr>
<td>Lifelong-seq</td>
<td>0.3140</td>
<td>0.2043</td>
<td>0.3447</td>
<td>0.2455</td>
</tr>
<tr>
<td>Lifelong-replay*</td>
<td>0.6225</td>
<td>0.5481</td>
<td>0.5485</td>
<td>0.4573</td>
</tr>
<tr>
<td>MeLL</td>
<td><strong>0.9379</strong></td>
<td><strong>0.9342</strong></td>
<td><strong>0.9271</strong></td>
<td><strong>0.9224</strong></td>
</tr>
</tbody>
</table>
Experiments (4)

✓ Ablation Study

- The meta knowledge plays an important role in overall model performance.

<table>
<thead>
<tr>
<th>Ablation</th>
<th>F1</th>
<th>Improv. Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>MeLL</td>
<td>0.9341</td>
<td>N/A</td>
</tr>
<tr>
<td>w/o Meta knowledge</td>
<td>0.9178</td>
<td>-1.63%</td>
</tr>
<tr>
<td>w/o Slow learner</td>
<td>0.9269</td>
<td>-0.72%</td>
</tr>
<tr>
<td>w/o LRU replacement policy</td>
<td>0.9380</td>
<td>+0.39%</td>
</tr>
</tbody>
</table>

✓ Parameter Analysis

(a) Accuracy w.r.t the learning rate of the slow learner.
(b) F1 w.r.t the learning rate of the slow learner.
(c) Accuracy w.r.t the learning rate of the fast learner.
(d) F1 w.r.t the learning rate of the fast learner.
Experiments (5)

✓ Online Deployment
  • A/B test on AliMe hotline system
  • Online system
    • Task-specific TextCNN models

✓ Salability Analysis
  • Number of parameters w. the number of tasks

<table>
<thead>
<tr>
<th>Method</th>
<th>F1</th>
<th>Relative Improv.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Online system (Single)</td>
<td>0.8359</td>
<td>N.A.</td>
</tr>
<tr>
<td>MeLL (w. LRU)</td>
<td>0.9079</td>
<td>8.61%</td>
</tr>
</tbody>
</table>
Conclusion

✓ We present the MeLL framework to address large-scale extensible user intent classification.

✓ Experiments and online A/B test show that MeLL consistently outperforms strong baselines.

✓ Future work:
  • How MeLL be employed to solve other tasks and support other applications.
THANKS

-------- Q&A Section --------