Customer Obsession Ticket Assistant

Improving Uber Customer Support with Natural Language Processing and Deep Learning

Piero Molino | AI Labs
Huaixiu Zheng | Applied Machine Learning
Yi-Chia Wang | Applied Machine Learning
Main Takeaways

COTA v1: classical NLP + ML models
- Faster and more accurate customer care experience
- Million $ of saving while retaining customer satisfaction

COTA v2: deep learning models
- Experiments with various deep learning architectures
- 20-30% performance boost compared to classical models
COTA Blog Post and followup, KDD paper

COTA: Improving Uber Customer Care with NLP & Machine Learning

By Huaixiu Zheng, Yi-Chia Wang, & Piero Molino

January 3, 2018
Motivation and Solution

Complexity of Customer support @Uber

COTA v1: Traditional ML / NLP Models

Multi-class Classification vs Ranking

COTA v2: Deep Learning Models

Deep learning architectures

COTA v1 vs COTA v2
Motivation and Solution

Complexity of Customer support @Uber

COTA v1: Traditional ML / NLP Models
  Multi-class Classification vs Ranking

COTA v2: Deep Learning Models
  Deep learning architectures

COTA v1 vs COTA v2
What is the challenge?
As Uber grows, so does our volume of support tickets

Millions of tickets from riders / drivers / eaters per week

Thousands of different types of issues users may encounter
What is the challenge?
And it is not easy to solve a ticket

1000+ types
in a hierarchy
depth: 3~6

10+ actions (adjust fare, add appeasement, …)

1000+ reply templates
Motivation and Solution

Complexity of Customer support @Uber

COTA v1: Traditional ML / NLP Models

Multi-class Classification vs Ranking

COTA v2: Deep Learning Models

Deep learning architectures

COTA v1 vs COTA v2
COTA v1: Suggested Resolution

Machine learning models recommending the 3 most relevant solutions

SUGGESTED CONTACT TYPES
- Driver > Account > Unable to sign in or go online > Account inactive
- Driver > Account > Profile > Unsubscribe > SMS or Text
- Driver > Account > Vehicles > Edit vehicle class

Reorder actions in relevance

Surface top-3 most-relevant reply templates
COTA v1 Model Pipeline

**FEATURES**
- User Information
- Trip Information
- Ticket Metadata
- Ticket Message
- Class Prototype

**PREPROCESSING**
- Tokenization
- Lowercasing
- Stopword removal
- Lemmatization
- TF-IDF
- LSA

**FEATURE ENGINEERING**
- Multiclass Classification

**ML ALGORITHM**
- Top 3 Contact Types
- Top 3 Reply Templates

**PREDICTIONS**

**A**
- Multiclass classification

**B**
- Pointwise ranking

Information about the contact type or reply, obtained from all tickets belonging to it
From Classification to Ranking

Multi-class Classification

<table>
<thead>
<tr>
<th>Tickets Features</th>
<th>Label (CT1, CT2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>t1 features</td>
<td>CT1</td>
</tr>
<tr>
<td>t2 features</td>
<td>CT2</td>
</tr>
</tbody>
</table>

Pointwise Ranking

<table>
<thead>
<tr>
<th>Tickets Features</th>
<th>Type Features</th>
<th>Sim (t, CT)</th>
<th>Label (0, 1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>t1 features</td>
<td>CT1 features</td>
<td>0.8</td>
<td>1</td>
</tr>
<tr>
<td>t1 features</td>
<td>CT2 features</td>
<td>0.1</td>
<td>0</td>
</tr>
<tr>
<td>t2 features</td>
<td>CT1 features</td>
<td>0.2</td>
<td>0</td>
</tr>
<tr>
<td>t2 features</td>
<td>CT2 features</td>
<td>0.7</td>
<td>1</td>
</tr>
</tbody>
</table>

Ranking allows us to include **features of candidate types** and **similarity features** between a ticket and a candidate type.

Model: **Random Forest** with hyperparameters optimized through **grid search**
Performance Comparison

6% absolute (10% relative) improvement

Hits@3: any of the top 3 suggestions is selected by CSRs
Motivation and Solution

Complexity of Customer support @Uber

COTA v1: Traditional ML / NLP Models

Multi-class Classification vs Ranking

COTA v2: Deep Learning Models

Deep learning architectures

COTA v1 vs COTA v2
COTA v2: Deep Learning Architecture

Generic architecture, reusable in many different applications. We are considering open-sourcing it!
COTA v2: Text Encoding Models

**Char CNN**
- Char Seq
- 6 x 1D Conv
- 2 x FC
- Vector

**Word CNN**
- Word Seq
- 1D Conv width 2
- 1D Conv width 3
- 1D Conv width 4
- 1D Conv width 5
- 2 x FC
- Vector

**Char / Word RNN**
- Char / Word Seq
- 2 x RNN
- 2 x FC
- Vector

**Char / Word CNN RNN**
- Char / Word Seq
- 3 x Conv
- 2 x RNN
- 2 x FC
- Vector
Which text encoder?

Hyperparameter search for contact type classification

<table>
<thead>
<tr>
<th>Model</th>
<th>Validation accuracy</th>
<th>Training time per epoch in minutes</th>
</tr>
</thead>
<tbody>
<tr>
<td>CharCNNRNN opt</td>
<td>0.4805</td>
<td>35</td>
</tr>
<tr>
<td>WordCNN opt</td>
<td>0.4733</td>
<td>4</td>
</tr>
<tr>
<td>WordRNN opt</td>
<td>0.4713</td>
<td>17</td>
</tr>
<tr>
<td>WordCNNRNN opt</td>
<td>0.4615</td>
<td>12</td>
</tr>
<tr>
<td>WordCNN</td>
<td>0.4598</td>
<td>5</td>
</tr>
<tr>
<td>CharCNN</td>
<td>0.4370</td>
<td>5</td>
</tr>
<tr>
<td>Random Forest</td>
<td>0.3920</td>
<td>5</td>
</tr>
</tbody>
</table>

WordCNN is the best compromise between performance and speed

20%+ over Random Forest used in COTA v1 and ~10x faster than CharCNNRNN
Sequence Model for Type Selection

Predict the sequence of nodes instead of leaf node

Example: Driver \textgreater Trips \textgreater Pickup and drop-off issues \textgreater Cancellation Fee \textgreater Driver Cancelled

Use a Recurrent Decoder to predict \textit{sequences of nodes} in the contact type tree.

Pick the last class before <eos> as prediction.

Model makes more reasonable mistakes.
Final Architecture

Multi-task sequential learning

- **Text features**
  - e.g. message
  - Convolution layers

- **Categorical features**
  - e.g. flow node
  - Embedding layer

- **Numerical features**
  - e.g. trip fare
  - Batch-norm layer

- **Binary features**
  - e.g. is completed

- **FC + Dropout layers**

- **Recurrent Decoder**

- **Softmax layer**

- **Train**
  - ground-truth

- **Test**
  - predicted
Effect of Adding Dependencies Between Tasks

Adding the dependency from Type to Reply improves accuracy.

It also improves a lot the coherence between the two models, increasing combined accuracy consistently.

Combined accuracy computed requiring both Type and Reply model to be correct at the same time.
Outline

Motivation and Solution

  Complexity of Customer support @Uber

COTA v1: Traditional ML / NLP Models

  Multi-class Classification vs Ranking

COTA v2: Deep Learning Models

  Deep learning architectures

COTA v1 vs COTA v2
COTA v2 is consistently more effective than COTA v1 on all metrics for both models.

The combined accuracy in particular shows an absolute ~+9% (relative ~+20%).
COTA v1 vs. COTA v2 A/B Test

Control

I had an issue with my pickup
Rider > Trips > Pickup and drop-off issues > Cancellation Fee

Search Contact Type

Cleaning fee -> Fare review
Cross Support - General -> Feedback about driver
Cross Support - Safety -> Feedback about vehicle
Duplicate contact -> Invoice
Info -> Lost items general info
Lost Items -> Pickup and drop-off issues
IRT: Accidents -> Promotions
IRT: Incidents -> Receipt
Service Denial -> uberPOOL on trip issues
Tech issues -> DOST
Trips -> External Sources

Brought to wrong destination -> Cancellation Fee
Cancelling fee -> Couldn’t find or get to driver
Had to walk to pickup or destination -> Driver arrived too early
No cars available -> Driver cancelled
Pickup difficulty without cancellation fee -> Driver didn’t answer phone
Scheduler rides -> Driver took too long
None of the above works

Treatment

I had an issue with my pickup
Rider > Trips > Pickup and drop-off issues > Cancellation Fee

Search Contact Type

Rider > Trips > Pickup and drop-off issues > Cancellation Fee > Driver cancelled
Rider > Trips > Pickup and drop-off issues > Cancellation Fee > Cancellation policy
Rider > Trips > Pickup and drop-off issues > Cancellation Fee > Couldn’t find or get to driver

DOST
External Sources -> Cancellation Fee
Fare review -> Couldn’t find or get to driver
Feedback about driver -> Driver arrived too early
Feedback about vehicle -> Driver cancelled
Invoice -> Driver didn’t answer phone
Lost Items general info -> Driver took too long
Lost Items general info -> Trips automatically cancelled
Pickup and drop-off issues -> Driver to do
Pickup and drop-off issues -> Driver to do
Pickup and drop-off issues -> Refused
Pickup and drop-off issues -> Road issue
Pickup and drop-off issues -> Set wrong
Pickup and drop-off issues -> Scheduled rides
Pickup and drop-off issues -> UberPool no show fee
UserPool automatically cancelled
UserPool no show fee
None of the above works
COTA v2 is 20-30% more accurate than COTA v1 in online A/B tests.

COTA v1 reduces handling time of ~8%, while COTA v2 provides an additional ~7% reduction, more than ~15% overall reduction.

Statistically significant customer satisfaction improvement.
Threshold on Type Model Confidence

TYPE model (accuracy=0.662)

Prediction Probability vs. # of Tickets

TYPE wordCNN Model

Data Coverage vs. $p_{acc}$, $p_{th}$
Threshold on Both Models’ Confidence
Coverage vs. Maximum Accuracy

95% accuracy → 10% coverage
90% accuracy → 20% coverage
Conclusions

Using NLP & ML COTA makes customer care experience **faster** and **more accurate** while saving Uber millions of $.

Moving from traditional to deep learning models, we observe a substantial **performance boost** (up to 30%).

Using intelligent suggestions we were able to **reduce ticket handling time** without impacting customer satisfaction.
COTA Team
Cross-functional collaboration

AI Labs
Applied Machine Learning
Customer Obsession
Michelangelo
Sensing and Perception