Deep Learning Applications in NLP and Conversational AI at Uber

Huaixiu Zheng

O’Reilly AI Conference
San Jose 2019
About Me

Physics

Quantum Computation & Quantum Physics
Phase Transition, Quantum Error Correction

Deep Learning
NLP & Conversational AI
Agenda

- Solve the right problem
- Find the right AI solution
- Put AI into the right system
“Successful problem solving requires finding the right solution to the right problem. We fail more often because we solve the wrong problem than because we get the wrong solution to the right problem.”

— Russell L. Ackoff
Solve the right problem
Customer Support

*Millions of tickets/week + thousands of issue types/solutions*

Large space of decision-making, error prone, low efficiency
In-App Text Communications

Painful to type messages on phones
Voice Interface

96.8%
Of drivers see texting while driving as a serious threat.

Cell phone use while driving is common. In the past month, 60.5% of drivers talked on a hands-free cell phone.

69%
Of drivers prefer hands-free phone while driving.

Drivers considered it acceptable to talk on a hands-free phone than hands-held phone (24.6%) while driving.

*Data retrieved from: AAA Foundation for Traffic Safety 2017 Traffic Safety Culture Index

Painful to tap on phones
Summary: the right problem

- Involving decision making under uncertainty
- Within the reach of AI technology
- Touching customer pain point
- Bringing business values
Find the right AI solution
What is the right AI solution?

- Traditional algorithms vs Deep Learning algorithms
- Unsupervised vs Supervised vs GAN vs RL
- CNN vs RNN vs Transformer
- Transfer learning vs training from scratch
- Modularized design vs end-to-end training
- ……
What is the right AI solution?
Data vs Model vs Task

Model Performance vs Data Size

- Large NN
- Medium NN
- Small NN
- Traditional ML

Data Size vs Task Complexity

Task Complexity
COTA: Customer Obsession Ticket Assistant
Customer Support Platform

- User
  - Select Flow Node
  - Write Message
- CSR
- Response
  - Select Contact Type
  - Lookup info & Policies
  - Select Action
  - Write response using a Reply Template
  - 1000+ templates
  - 50+ actions
- Contact Ticket
- 1000+ types
Data vs Model vs Task

Model Performance vs Data Size:
- Large NN
- Medium NN
- Small NN
- Traditional ML

Data Size vs Task Complexity:
- Linear relationship with increasing complexity.
COTA v1: Pointwise Ranking

AB Test: ~10% Handle Time Reduction
COTA v2: Deep Learning Architecture

**Input Encoders**
- Text features
- Categorical features
- Numerical features
- Binary features
- Set features
- Sequential features

**Combiner**

**Output Decoders**
- Text features
- Categorical features
- Numerical features
- Binary features
- Set features
- Sequential features

*Generic architecture, reusable* in many different applications
COTA v1 vs. COTA v2: Type and Reply

- COTA v2 is **consistently more effective** than COTA v1 on all metrics for both models
- Combined accuracy: an absolute +7% (relative +15%)
Data vs Model vs Task

Model Performance vs Data Size

- Large NN
- Medium NN
- Small NN
- Traditional ML

COTA

Data Size vs Task Complexity

COTA
OCC: One-Click Chat
Smart Replies for Drivers
Challenges

“Where Are u”

“Where are you going” <> “Where are you heading to”

“I am coming” <> “I am here”

"im washington for you"
Data vs Model vs Task

![Graph showing data size vs task complexity with OCC and Gmail as examples.](image)
Two-Step Algorithm

Key: Semantic Understanding

Intent Detection

Incoming Message → Intent Detection → Reply Retrieval

“Where are you right now?”

Reply Retrieval

Response Template 1: Yes, I am on my way
Response Template 2: I am stuck in traffic
Response Template 3: I am here

I am here

I am stuck in traffic

Yes, I am omw
Intent Detection
Semantic Understanding

Unsupervised Learning
1. Train Doc2vec model: message $\rightarrow$ dense vector
2. Map Labeled data to embedding space
3. Calculate centroid of each intent cluster
4. Distance between incoming message and intent cluster centroid
5. Classify into NN Intent Cluster

“Where are you right now?”

Input

Doc2Vec

Output

[0.6389473, 0.1192103, ....., 0.8931793719]
Intent Detection
Use labeled data

Message Encoding
1. Train Doc2vec model: message -> dense vector
2. Map Labeled data to embedding space
3. Calculate centroid of each intent cluster
4. Distance between incoming message and intent cluster centroid
5. Classify into NN Intent Cluster

Thousands of labeled data
Intent Detection: Nearest Neighbor Classifier

**Intent Cluster Centroid**
1. Train Doc2vec model: message -> dense vector
2. Map Labeled data to embedding space
3. Calculate centroid of each intent cluster
4. Distance between incoming message and intent cluster centroid
5. Classify into NN Intent Cluster
Intent Detection

Handle new incoming message

Distance to Intent Clusters

1. Train Doc2vec model: message -> dense vector
2. Map Labeled data to embedding space
3. Calculate centroid of each intent cluster
4. Distance between incoming message and intent cluster centroid
5. Classify into NN Intent Cluster
Intent Detection
Incoming message → Intent

Nearest Neighbor Classifier (NNC)
1. Train Doc2vec model: message → dense vector
2. Map Labeled data to embedding space
3. Calculate centroid of each intent cluster
4. Distance between incoming message and intent cluster centroid
5. Classify into Nearest Neighbor Intent Cluster

"im washington for you" → "I’m waiting for you"
Intent Detection: NNC vs Deep Learning

Incoming message → Intent

Top-1 Accuracy

0.155

NNC = Nearest-Neighbor Classifier
Data vs Model vs Task

Model Performance

Data Size

OCC     COTA

Large NN
Medium NN
Small NN
Traditional ML

Data Size

Task Complexity

OCC     COTA
Scenario 1

Dispatch Alert
Scenario 2

Pick-up Communication

Your response

I will be right there.
Neural Language Correction
Sentence-level Ranking

Data Size

Task Complexity

SLR
Word-level Ranking

Data Size

SLR  WLR

Task Complexity

Word Index

Tag

O  O  B  I  I  I

RNN

Concatenate embeddings

Embedding Layer

N-Best

oscar  *  which  side  you  come

oscar  *  which  side  you  came

ask  her  which  side  you  came

Timesteps (T)
Word-level Generation
Performance

Message Error Rate Reduction

Data Size

Task Complexity

SLR  WLR  WLG

WLG
Summary: the right AI solution

- Complexity of the Task
- Amount of Data Available
Put AI into the **right** system
Perception

ML

Code
Reality

Sculley et al., NIPS (2015)
Challenges

- **System Dependencies**
- Data Dependencies
- Feedback Loops
- Dynamics in External World
- Model Interpretability
- ......
Traditional vs Deep Learning Stacks

* TensorFlow, the TensorFlow logo and any related marks are trademarks of Google Inc.
* Core i7 is a trademark of Intel Corporation or its subsidiaries in the U.S. and/or other countries.
* NVIDIA and GeForce are registered trademarks of NVIDIA Corporation.
DL Production: Deep Learning Spark Pipeline

Spark + Tensorflow
Serving: Apache Spark + JVM for both batch and real-time requests
Summary

- Right problem: decision-making, within AI reach, pain point, biz values
- Right AI solution: task complexity vs data size vs model complexity
- Right system: infrastructure-aware AI ecosystem
Acknowledgments

Conversation AI team
Uber AI Labs
Michelangelo team
Customer Support team
Communication Platform team
Product/UX Design team
Thanks!
COTA v2: Text Encoding Models

Word CNN

- Word Seq
- 1D Conv width 2
- 6 x 1D Conv
- 2 x FC
- Softmax

Char CNN

- Char Seq
- 6 x 1D Conv
- 2 x FC
- Softmax

Char / Word RNN

- Char / Word Seq
- 2 x RNN
- 2 x FC
- Softmax

Char / Word CNN RNN

- Char / Word Seq
- 3 x Conv
- 2 x RNN
- 2 x FC
- Softmax