Stay calm, we'll deal with the problem... training chatbots from customer service interactions

Benoit Favre

Laboratoire d'Informatique Fondamentale

February 24, 2017

What is a chatbot?

- Dialog system which can have an entertaining conversation
  - Chat-chat
  - Task oriented
- History
  - Eliza, virtual therapist
    - `http://www.masswerk.at/elizabot/`
  - Mitsuku (best chatbot at Loebner price 2013/2016)
    - `http://www.mitsuku.com/`
  - The Microsoft Tay fiasco
    - Humans will always try to defeat an IA
  - A new industry hype
    - Facebook, google...
- Question: can we spare dialog model engineering?
  - Train a model directly from conversation traces

Motivation

- Datcha project
  - `http://datcha.lif.univ-mrs.fr`
  - Study text chat NLP
  - Relationship between argumentation and semantics
  - Task-oriented evaluation
- Data-driven approach to conversation modeling
  - Given a conversation up to a point, can we predict what will happen next
  - No need for linguistic analysis, but no linguistic prior
- “Chatbot" if we predict the next utterance for one of the parties
  - Simulate an agent to solve simple customer problems
  - Simulate a customer to train agents
  - However, we cannot actionate anything as those events have not been logged
- If we can create a chatbot, we have understood something about conversations
  - May not be an objective of the project
  - But could impact the way we think about conversations

Related work

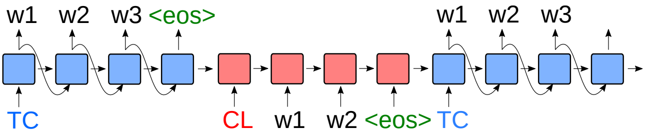
- Models
  - Generate next turn given previous turn with an encoder-decoder
    - "A Neural Conversational Model" [Vynials et al. 2015]
  - Add turn-level representations
    - "Building End-To-End Dialogue Systems Using Generative Hierarchical Neural Network Models" [Serban et al., AAAI 2016]
  - Add attention mechanism to the hiearchical model
    - "Attention with Intention for a Neural Network Conversation Model" [Yao et al., SLUNIPS-2015]
  - Chatbot as information retrieval
    - "Improved Deep Learning Baselines for Ubuntu Corpus Dialogs" [Kadlec et al., SLUNIPS-2015]
- Dialog specifics
  - Introduce long term reward
    - "Deep Reinforcement Learning for Dialogue Generation", [Li et al., ACL 2016]
  - How generate diverse responses?
    - "A Diversity-Promoting Objective Function for Neural Conversation Models" [Li et al., NAACL 2016]
  - Enforce consistency by explicitly modeling speakers
    - "A Persona-Based Neural Conversation Model" [Li et al., ACL 2016]
- Evaluation: automatic metrics do not correlate with manual evaluation
  - "How NOT To Evaluate Your Dialogue System" [Liu et al, EMNLP 2016]

Proposed models

- Our approach
  - Knowledge-free and structure-free
  - Learn directly from data in an end-to-end manner
  - Deep learning to the rescue
- Two relatively naive models
  1. Alternating language models
     - Essentially a language model
  2. Turn retrieval
     - Predict a complete turn at a time

Model 1: Alternating LM + LSTM

- A simplified version of the encoder-decoder (or seq2seq) framework
  - Trained the same way as a regular word-based language model
  - At prediction time, alternate between user input and generation
    - ★ Training data needs to be in the same form
- Current implement: multi-layer LSTM



LM: Math

- LSTM

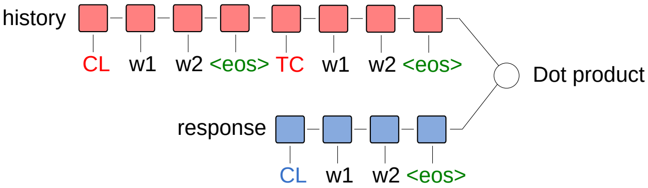
$$
\begin{aligned}
i_t &= \sigma(W_i x_t + U_i h_t + b_i) && (1)\\
f_t &= \sigma(W_f x_t + U_f h_t + b_f) && (2)\\
o_t &= \sigma(W_o x_t + U_o h_t + b_o) && (3)\\
c'_t &= \tanh(W_c x_t + U_c h_t + b_c) && (4)\\
c_{t+1} &= f_t \odot c_t + i_t \odot c'_t && (5)\\
h_{t+1} &= o_t \odot \tanh(c_{t+1}) && (6)\\
\mathrm{LSTM}(x_t, h_t, c_t) &= h_{t+1} && (7)
\end{aligned}
$$

- Multilayer LSTM language model

$$
\begin{aligned}
h_0^{(i)} &= c_0^{(i)} = 0 \quad \forall i \in [1; n] && (8)\\
x_t &= embedding(w_t) && (9)\\
h_{t+1}^{(1)}, c_{t+1}^{(1)} &= LSTM(x_t, h_t^{(1)}, c_t^{(1)}) && (10)\\
h_{t+1}^{(i)}, c_{t+1}^{(i)} &= LSTM(h_t^{(i-1)}, h_t^{(i)}, c_t^{(i)}) \quad \forall i \in]1; n] && (11)\\
LM(w_{t+1}) &= softmax(W_d h_{t+1}^{(n)} + b_d) && (12)
\end{aligned}
$$

Model 2: Bi-encoder GRU

- Create an information retrieval system
  - Which can retrieve the next turn given a history
  - Encode history with a first recurrent model
  - Encode next turn with a second recurrent model
  - Compute a similarity between those representations (dot product)
- Training objective: triplet ranking
  - Make sure the correct association has a higher score than a randomly selected pair
- Problem: the cost of retrieving a turn
  - Everything can be precomputed, just the dot product remains
  - Many approaches for finding approximate nearest neighbors in a high dimensional space (ie. locality preserving hashing)

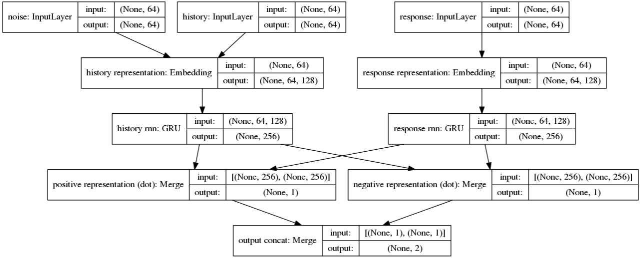


Bi-encoder training

- Maximize margin between the result of $h_i \cdot r_i$ and $n_i \cdot r_i$
  - h_i is the history
  - n_i is a random history
  - r_i is the response

$$Loss = \frac{1}{n} \sum_i \max(0, 1 - h_i \cdot r_i + n_i \cdot r_i))$$

- Keras model



GRU Math

- GRU

$$
\begin{aligned}
z_t &= \sigma(W_z x_t + U_z s_t + b_z) && (13)\\
r_t &= \sigma(W_r x_t + U_r s_t + b_r) && (14)\\
h_t &= \tanh(W_h x_t + U_h(r_t \odot s_t) + b_h) && (15)\\
s_{t+1} &= (1 - z_t) \odot h_t + z_t \odot s_t && (16)\\
\mathrm{GRU}(s_t, x_t) &= s_{t+1} && (17)
\end{aligned}
$$

- Bi-encodeur

$$
\begin{aligned}
x_{h,t} &= embedding(w_{h,t}) \quad \forall t \in [0, n] && (18)\\
x_{r,t} &= embedding(w_{r,t}) \quad \forall t \in [0, m] && (19)\\
h_i &= GRU_h(...GRU_h(0, x_{h,0}), ...x_{h,n}) && (20)\\
r_i &= GRU_r(...GRU_r(0, x_{r,0}), ...x_{r,m}) && (21)\\
BI\_ENC(h_i, r_i) &= softmax(h_i \cdot r_i) && (22)
\end{aligned}
$$

Evaluation setup

- Corpus: Orange ATH TV

Stat	Train	Valid	Test
Conversations	16,140	698	606
Turns	465,693	20,090	18,392
Words	7,744,262	327,979	299,340

- Preprocessing
  - Tokenization (based on penn tokenizer)
  - A few rules to strip additional URLs, phone numbers, etc.
  - Lower case
  - Concatenate turns of the same participant with `<eol>`
  - Separate conversations by `<eoc>`
  - Replace all `TC[1-9]` by a generic `TC`

Experiments

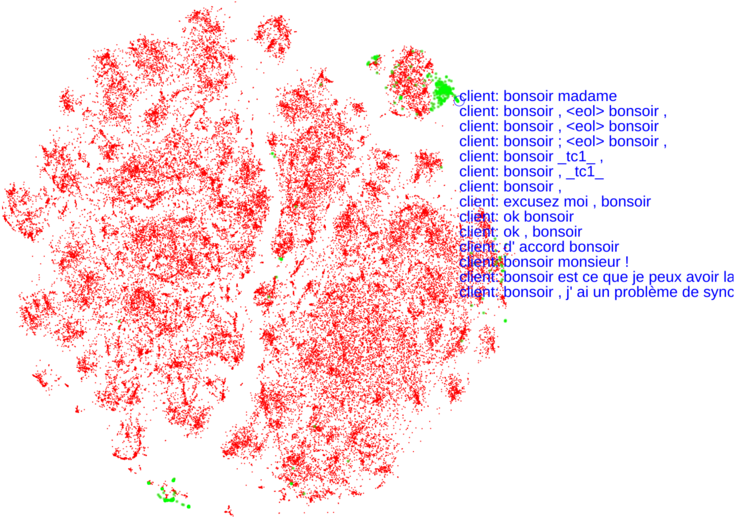
- Evaluation metrics
  - Perplexity (PPL): $-\frac{1}{n}\sum logP(turn|history)$
  - Better-than-random (BTR): $\frac{1}{n}|P(turn|history) > P(turn|noise)|$
- Results on the ATH TV test set (3 last files):

Method	PPL	BTR
Language model	17.52	69.39%
Information retrieval	11.85	93.91%

- Parameters
  - LM: vocab=30k, layers=2, hidden=650, sample=1024, maxlen=35, batch=20, optim=sgd, epochs=8
  - Bi-encoder: vocab=30k, embeddings=128 (init=w2v), hidden=256, maxlen=64, repr=128, batch=256, optim=Nadam, epochs=100

Analysis

- t-SNE Projections of turn representations



Prospects

- Better chatting
  - Look into attentive and memory-based models
    - Track entities
  - Stronger LM, better sampling in retrieval method
    - Introduction of reinforcement learning
  - Evaluation methodology
- Representations
  - Split conversation and use one side to predict the other
    - Look at trajectories in that space
  - Identify dialog acts / or dialogic structures in embeddings
- Applications to Datcha-relevant problems
  - Recurrent modeling for success prediction
  - Application to in-call agent tutoring