Sentences Generation

2016.05.11

陈云川
Outline


Motivation

- Lots of applications rely on text generation
  - speech recognition
  - machine translation
  - text rephrasing
  - question answering
Sequence Generation Model

\[ p(y_1, y_2, \ldots, y_m \mid x_1, x_2, \ldots, x_n) = \prod_{t=1}^{m} p(y_t \mid D_{t-1}) \]

\[
D_{t-1} = \{v_{t-1}\} \\
u = f(x_1, x_2, \ldots, x_n) \\
v_0 = I(u) \\
v_t = g(v_{t-1})
\]

\[
D_{t-1} = \{u, v_{t-1}\} \\
u = f(x_1, x_2, \ldots, x_n) \\
v_t = g(y_1, y_2, \ldots, y_t)
\]

Sequence to Sequence Learning
with Neural Networks

NIPS '14

ICLR '15
Neural Machine Translation
by Jointly Learning to Align and Translate

Dzmitry Bahdanau
Jacobs University Bremen, Germany

KyungHyun Cho
Université de Montréal

Ilya Sutskever
Google
ilyasu@google.com

Oriol Vinyals
Google
vinyals@google.com

Quoc V. Le
Google
qvl@google.com

Joshua Bengio*

Search to Decode

• How to sample a sequence from $\prod_{t=1}^{m} p(y_t | D_{t-1})$?

• Prefix search

• Beam search

• ???
Other Way to Decode?

- Prefix searching is too expensive
- Not pleased with the approximate beam searching
- Is it possible to generate a coarse sequence first and then refine it iteratively?

Click here to read more than the New York Times
Click here to read more from the New York Times
Generating Sequence with Deep Q-Network: the Model

- Generating the state Representation with LSTM
  - Encode
  - Decode
- Iteratively Decoding Sequence with Deep Q-Network (DQN)
Decoding Sequence with Deep Q-Network (DQN)

- Markov Decision Process
- States: (EnSent, DeSent)
- Actions: words and their positions
- State Transition Prob.: <Deterministic>
- Reward: BLEU score

Loss: \[ L_i(\theta) = \mathbb{E}_{s,a}[(q_i - Q(s, a; \theta_i))^2] \]

\[ q_i = \mathbb{E}_{s,a}[r_i + \lambda \max_{a'} Q(s', a'; \theta_{i-1})] \]
Empirical Observations on Model Design

- Separating Make State Generation Function from DQN
- Pre-training the State Generation Function
- Updating with Replay Memory Sampling
- Importance of Supervised Softmax Signal
Experimental Result

<table>
<thead>
<tr>
<th>Testing systems</th>
<th>LSTM decoder</th>
<th>DQN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average SmoothedBLEU on sentences IN the training set</td>
<td>0.425</td>
<td>0.494</td>
</tr>
<tr>
<td>Average SmoothedBLEU on sentences NOT in the training set</td>
<td>0.107</td>
<td>0.228</td>
</tr>
</tbody>
</table>
Connectionist temporal classification (CTC)
Connectionist Temporal Classification (CTC)

\[ p(\pi | x) = \prod_{t=1}^{T} y_{\pi_t}^{t} \]

\[ p(l | x) = \sum_{\pi \in B^{-1}(l)} p(\pi | x) \]

\( B \) is the many-to-one map that remove all repeated symbols and blanks

\[ B(\emptyset a a \emptyset \emptyset a b b) = a a b \]
Forward Variable

- the forward variable $\alpha$ is the summed probability of all length $t$ paths that are mapped by $\mathcal{B}$ onto the length $u/2$ prefix of $l$

$$\alpha(t, u) = \sum_{\pi \in V(t, u)} \prod_{i=1}^{t} y_{\pi_i}^i$$

$$\alpha(t, u) = y_{l_u}^t \sum_{i=f(u)}^{u} \alpha(t-1, i)$$

$$f(u) = \begin{cases} 
    u - 1 & \text{if } l'_u = \text{blank or } l'_{u-2} = l'_u \\
    u - 2 & \text{otherwise}
\end{cases}$$

$$V(t, u) = \{ \pi \in L^t : \mathcal{B}(\pi) = l_{1:u/2}, \pi_t = l'_u \}$$
Backword Variable

- the backward variable $\beta(t, u)$ is defined as the summed probabilities of all paths starting at $t+1$ that complete $l$ when append to any path contributing to $\alpha(t, u)$

$$\beta(t, u) = \sum_{\pi \in W(t, u)} \prod_{i=1}^{T-t} y_{\pi_i}^{t+i}$$

$$W(t, u) = \{ \pi \in L^{T-t} : B(\hat{\pi} + \pi) = l, \forall \hat{\pi} \in V(t, u) \}$$
\[ \mathcal{L}(S) = -\ln \prod_{(x,z) \in S} p(z|x) \]

\[ p(z|x) = -\sum_{(x,z) \in S} \ln p(z|x) \]

\[ \mathcal{L}(x, z) = -\ln \sum_{t=1}^{T} \sum_{u=1}^{n} \alpha(t,u) \beta(t,u) \]

\[ \mathcal{L}(x, z) = -\ln \left( \sum_{t=1}^{T} \sum_{u=1}^{n} \alpha(t,u) \beta(t,u) \right) \]
Loss Gradient

\[ \frac{\partial \mathcal{L}(x, z)}{\partial y^t_k} = -\frac{\partial \ln p(z|x)}{\partial y^t_k} = -\frac{1}{p(z|x)} \frac{\partial p(z|x)}{\partial y^t_k} \]

\[ \frac{\partial p(z|x)}{\partial y^t_k} = \frac{1}{y^t_k} \sum_{u \in B(z, k)} \alpha(t, u) \beta(t, u) \]

\[ \therefore p(z|x) = \sum_{u=1}^{\left|z'\right|} \alpha(t, u) \beta(t, u) \]
Decoding

\[ l^* = \arg \max_l p(l | x) \]

- Best Path Decoding
  \[ l^* \approx B(\pi^*) \]

- Prefix Search Decoding
Experimental Results
Tanks!