4C1-2

# 深層応答

## Attention-based Response Generation using Batch Normalization

坂原誠 *1　岡田将吾 *1　新田克己 *1
Makoto Sakahara　Shogo Okada　Katsumi Nitta

*1東京工業大学
Tokyo Institute of Technology

We propose a novel neural responding machine. In this model, batch normalization is applied to attention mechanism to help optimization. We also build a system can collect enormous amount of conversational tweet for training the model. Experiments result show that the model achieves lower perplexity with at least 1.4 times faster in training time, and outperforms a state-of-the-art attention-based model. Furthermore, qualitative analysis reveals that the model can generate natural responses to unseen tweet.

## 1. 序論

人工知能の研究領域において, 深層学習を用いた end-to-end で訓練されたモデル [Bahdanau 14, Xu 15, Amodei 15] が, 従来は困難とされてきた問題に対して, 一定の成果を収め, 専門知識にもとづいた人手による特徴量設計を組み合わせた古典的なモデルを置き換えつつある. これには, 従来の機械学習の手法に対して, 深層学習はデータ量とパラメータ数双方に対してスケールする性質をもつ, という背景が存在する.

オープドメインな会話の理解もまた, 人工知能分野における最も困難な問題のひとつとして広く知られている. しかし, 幸いなことに, Twitter *1 をはじめとするマイクロブログサービスの台頭により, 会話データは激増している. このことから, 深層学習により, 単語の概念と会話の応答規則を同時に学習するための手法 [Sordoni 15, Wen 15, Shang 15] が提案されている. しかし, ここには幾つかの問題が存在している.

まず, 会話データは膨大に存在すると考えられているが, 実際には, 研究者がそれを利用するための手段は乏しい. 先行研究では, クローズドコミニティから特定のユーザに関してサンプリングされた会話データ [Ritter 11, Shang 15], あるいは, 少量の特定タスクの対話コーパス [Sordoni 15, Wen 15] を用いて学習を行っている. また, 一般的に, 実データを用いて深いニューラルネットワーク (DNN) を訓練することは, 長時間の訓練と困難な最適化を必要とする.

これらの問題に取り組むために, 我々は, 一般利用可能な API のみを用いて, リアルタイムに, 会話ツイートを大規模収集するシステム *2 を開発した. また, 収集された会話ツイートの学習に対して, 安定的な最適化と高速な訓練を行える batch normalized attention-based neural responding machine を提案する.

## 2. 会話ツイートの大規模収集

statuses/sample によりツイートデータのグローバルストリームからサンプリングされたツイートがリプライであるとき, ツイートオブジェクトをエンキューする. ツイートオブジェクトは非同期にデキューされ, statuses/lookup により 100 リプライを同時にリプライ先に向かってトラバースする. 葉ノードに到達したシーケンスは, 会話オブジェクトとしてデータベース (DB) に格納される. ここで, リプライ先が Least Recently Used にマッチした場合, トラバース中のシーケンス, あるいは直接 DB に格納された会話オブジェクトに結合される. したがって, 現在進行中のリプライも含めてひとつのユニークな会話として処理される.

DNN をより良く訓練するためには, データの継続的な大規模収集は必要不可欠である. 我々の知る限り, 不特定多数の取得できうる限りの会話ツイートを全て処理可能 *3 なシステムは公開されていない. サンプリングされたツイート数が増加するにつれて, その統計的性質が, グローバルストリームに近づくことが指摘されている [Morstatter 13] .

## 3. アテンションメカニズムにもとづくニューラルネットによる応答生成

本研究では, attention mechanism[Bahdanau 14] にもとづいた neural responding machine (NRM) をベースラインとする.

エンコーダは, 入力シーケンス $\boldsymbol{x} = (x_1, \cdots, x_S)$ をアノテーションシーケンス $\boldsymbol{h} = (h_1, \cdots, h_S)$ に変換する. そして, デコーダは, 各時間ステップ $t$ において適切に重み付けられたアノテーションシーケンス $\boldsymbol{h}$ に着目することで, 出力シーケンス $\boldsymbol{y} = (y_1, \cdots, y_T)$ を生成する. エンコーダとデコーダ双方には, recurrent neural networks (RNNs) が用いられる.

アフィン変換 $y = Wx + b$ を $\mathcal{T}_{p,q} : \mathbb{R}^p \to \mathbb{R}^q$ とする. ここで, $W$, $x$, $y$, $b$ はそれぞれ重み行列, 層への入力, 出力とバイアスベクトルとする. さらに, 要素ごとのある非線形変換 $\phi$ を伴うアフィン変換を $\mathcal{F}_\phi(\cdot)$ とする.

### 3.1 エンコーダ

エンコーダは bidirectional RNN (BiRNN, [Schuster 97]) によって表現される. BiRNN は, 前向き $\vec{\boldsymbol{h}} = (\vec{h_1}, \cdots, \vec{h_S})$ と後ろ向き $\overleftarrow{\boldsymbol{h}} = (\overleftarrow{h_1}, \cdots, \overleftarrow{h_S})$ 双方向の隠れ層を結合することで, アノテーションシーケンス $\boldsymbol{h} = [\vec{\boldsymbol{h}}, \overleftarrow{\boldsymbol{h}}]^{\mathrm{T}}$ を求める. アノテーションシーケンスは, 各時間ステップ $s$ において, 入力シーケンス $\boldsymbol{x}$ 全体の情報を異なるダイナミクスで保持する.

Contact: Makoto Sakahara,
sakahara@ntt.dis.titech.ac.jp, makoto.sakahara@gmail.com,
https://github.com/makotosakahara

*1 https://twitter.com/

*2 第一著者の GitHub にて公開予定

*3 2015 年 8 月中旬より 2016 年 3 月現在に至るまで, 英語および日本語会話ツイートの収集システムを継続稼働中

シーケンス中の複雑なパターンを捉えるために, RNNs の活性化関数には, long short-term memory (LSTM, [Hochreiter 97]) や gated recurrent unit (GRU, [Cho 14]) が典型的に用いられる. LSTM を用いて, エンコーダ は次式で表される.

$$\begin{pmatrix} i_s \\ f_s \\ o_s \\ \tilde{m}_s \end{pmatrix} = \begin{pmatrix} \sigma \\ \sigma \\ \sigma \\ \tanh \end{pmatrix} \mathcal{T}_{l+n,n} \begin{pmatrix} \acute{\boldsymbol{E}} x_s \\ h_{s-1} \end{pmatrix} \quad (1)$$

$$m_s = f_s \odot m_{s-1} + i_s \odot \tilde{m}_s \quad (2)$$

$$h_s = o_s \odot \tanh(m_s) \quad (3)$$

ここで, $i_s$, $f_s$, $o_s$, $\tilde{m}_s$, $m_s$, $h_s$ はそれぞれ LSTM の入力門, 忘却門, 出力門, 記憶素子候補, 記憶素子と隠れ層である. $l$, $n$ はそれぞれ単語埋め込みと隠れ層の次元数, そして, $\acute{\boldsymbol{E}} \in \mathbb{R}^{l \times K}$ は語彙サイズ $K$ のエンコーダの単語埋め込み行列である. また, $\sigma$ と $\odot$ はそれぞれロジスティックシグモイド関数と要素積である.

### 3.2 デコーダ

デコーダは, 1 ステップ前の隠れ層 $h_{t-1}$, 1 ステップ前に生成されたトークン $y_{t-1}$, そして重み付けられたアノテーションシーケンス $\boldsymbol{h}$ の総和であるコンテキストベクトル $z_t$ にもとづいた隠れ層 $h_t$ を計算することで, トークン $y_t$ を生成する. デコーダは次式で表される.

$$\begin{pmatrix} i_t \\ f_t \\ o_t \\ \tilde{m}_t \end{pmatrix} = \begin{pmatrix} \sigma \\ \sigma \\ \sigma \\ \tanh \end{pmatrix} \mathcal{T}_{l+n+2n,n} \begin{pmatrix} \grave{\boldsymbol{E}} y_{t-1} \\ h_{t-1} \\ z_t \end{pmatrix} \quad (4)$$

$$m_t = f_t \odot m_{t-1} + i_t \odot \tilde{m}_t \quad (5)$$

$$h_t = o_t \odot \tanh(m_t) \quad (6)$$

ここで, $\grave{\boldsymbol{E}}$ は デコーダの単語埋め込み行列である. コンテキストベクトル $z_t$ は時間ステップ $t$ ごとに次式で計算される.

$$z_t = \sum_{s=1}^{S} \alpha_{ts} h_s \quad (7)$$

$$\alpha_{t\cdot} = \mathcal{F}_{\text{softmax}}(e_{t\cdot}) \quad (8)$$

$$e_{ts} = \mathcal{F}_{\tanh}(h_{t-1}, h_s) \quad (9)$$

時間ステップ $t$ における各アノテーション $h_s$ の重要度を表すエネルギー $e_{ts}$ を, softmax 関数で正規化することで, アノテーションシーケンス $\boldsymbol{h}$ に対する重み $\alpha_t$ を得る. $\alpha_{ts}$ は, 入力 $x_s$ が $y_t$ に変換される, あるいは関係する確率として解釈できる.

また, デコーダの初期状態は, アノテーションシーケンス $\boldsymbol{h}$ の算術平均を用いて, 次式で与えられる.

$$m_0 = \mathcal{F}_{\tanh}(\frac{1}{S}\sum_{s}^{S} h_s) \quad (10)$$

$$h_0 = \mathcal{F}_{\tanh}(\frac{1}{S}\sum_{s}^{S} h_s) \quad (11)$$

そして, 生成されるトークン $y_t$ の確率は, 隠れ層 $h_t$, 1 ステップ前に生成されたトークン $y_{t-1}$, コンテキストベクトル $z_t$ の条件付き確率で与えられる.

$$p(y_t|h_t, y_{t-1}, z_t) = \mathcal{F}_{\text{softmax}}(\mathcal{D}(\mathcal{F}_{\text{maxout}}(h_t, y_{t-1}, z_t))) \quad (12)$$

ここで, $\mathcal{D}$ は Dropout[Srivastava 14] 演算子である. readout 層の活性化関数には, 区分線形関数である maxout 関数 [Goodfellow 13] を用いる.

## 4. RNNs のためのバッチ正規化

DNN において, 訓練中に各層への入力分布が変化すると, 各層はそのつど新しい分布に適応しようとする. その結果, ネットワーク全体としての最適化が妨げられることになる. batch normalization [Ioffe 15] は, *internal covariate shift* と呼ばれるこの分布の変化を削減する.

$m$ 個のサンプルからなるミニバッチ $\mathcal{B}$ が与えられたとき, バッチ軸に沿った $k$ 個の特徴量の標本平均と標本分散は, 次式で計算される.

$$\overline{\mathcal{B}}_k = \frac{1}{m}\sum_{i=1}^{m} \mathcal{B}_{ik} \quad (13)$$

$$\sigma_k^2 = \frac{1}{m}\sum_{i=1}^{m} (\mathcal{B}_{ik} - \overline{\mathcal{B}}_k) \quad (14)$$

各特徴量は, このバッチ統計情報を用いて, 次式で正規化される.

$$\hat{\mathcal{B}}_k = \frac{\mathcal{B}_k - \overline{\mathcal{B}}_k}{\sqrt{\sigma_k^2 + \epsilon}} \quad (15)$$

ここで $\epsilon$ は, 数値的安定性のための小さな正の定数である.

しかしながら, 単純に正規化するだけでは, 各層は非線形性による表現能力を失ってしまう. そこで, 表現能力を復元できるように, 学習可能なスケールパラメータ $\gamma$ とシフトパラメータ $\beta$ を導入する.

$$\mathcal{BN}(\mathcal{B}_k) = \gamma_k \hat{\mathcal{B}}_k + \beta_k \quad (16)$$

ここで, $\mathcal{BN}$ は Batch normalization 演算子である. 要素ごとのある非線形変換を伴うアフィン変換 $\mathcal{F}_\phi(\cdot)$ に対して, $\mathcal{BN}$ は次のように適用される. ここで, バイアスベクトルの影響は, 正規化により無効化されるので, バイアス項は取り除かれる.

$$\begin{aligned} y &= \phi(\mathcal{BN}(Wx)) \\ &= \phi(\mathcal{T}(\mathcal{BN}(x))) \\ &= \mathcal{F}_\phi(\mathcal{BN}(x)) \end{aligned} \quad (17)$$

テスト時には, バッチ統計情報を求めることができないので, 代わりに, 訓練時の移動平均を用いて正規化する.

一般的に, RNNs は可変長が固定長にパディングされたミニバッチを扱うので, (13, 14) 式でバッチ統計情報を求めることができない. 代わりに, バッチ軸と時間軸双方に沿って, 各特徴量の標本平均と標本分散を求める. これは, sequence-wise normalization[Laurent 15] と呼ばれる.

$$\overline{\mathcal{B}}_k = \frac{1}{n}\sum_{i=1}^{m}\sum_{t=1}^{T} \mathcal{B}_{itk} \quad (18)$$

$$\sigma_k^2 = \frac{1}{n}\sum_{i=1}^{m}\sum_{t=1}^{T} (\mathcal{B}_{itk} - \overline{\mathcal{B}}_k) \quad (19)$$

ここで, $n$ と $T$ はそれぞれ非パディングトークンの総数と各シーケンス長である.

## 5. バッチ正規化を適用したアテンションメカニズム

我々は,batch normalization を attention mechanism に対して適用する. エンコーダとデコーダ双方の LSTM に対して, $\mathcal{BN}$

は時間ステップ $t$ に依存しない非再帰結合のみに適用する.

$$\begin{pmatrix} i_s \\ f_s \\ o_s \\ \tilde{m}_s \end{pmatrix} = \begin{pmatrix} \sigma \\ \sigma \\ \sigma \\ \tanh \end{pmatrix} \mathcal{T}_{l+n,n} \begin{pmatrix} \mathcal{BN}\left(\acute{\boldsymbol{E}} x_s\right) \\ h_{s-1} \end{pmatrix} \tag{20}$$

$$\begin{pmatrix} i_t \\ f_t \\ o_t \\ \tilde{m}_t \end{pmatrix} = \begin{pmatrix} \sigma \\ \sigma \\ \sigma \\ \tanh \end{pmatrix} \mathcal{T}_{l+n+2n,n} \begin{pmatrix} \mathcal{BN}\left(\grave{\boldsymbol{E}} y_{t-1}\right) \\ h_{t-1} \\ z_t \end{pmatrix} \tag{21}$$

同様に,(9) 式に $\mathcal{BN}$ を適用する.

$$e_{ts} = \mathcal{F}_{\tanh}(h_{t-1}, \mathcal{BN}(h_s)) \tag{22}$$

readout 層に対しては, $(h_t, y_{t-1}, z_t)$ の線形結合に対して, $\mathcal{BN}$ を適用する. ただし, 非線形関数が maxout であるとき, $\mathcal{BN}$ は, maxout のグループごとに適用する必要性があることが実験的に判明した.

$$p(y_t|h_t, y_{t-1}, z_t) = \mathcal{F}_{\text{softmax}}(\mathcal{F}_{\text{maxout}}(\mathcal{BN}(h_t, y_{t-1}, z_t))) \tag{23}$$

したがって, 提案する attention mechanism は, ネットワーク全体を通して, 層間の結合がバッチ正規化されている. このとき Dropout $\mathcal{D}$ を適用する必要性はない.

## 6. 実験

我々は, (4) 式からコンテキストベクトル項を除いた no-attention NRM, attention-based NRM, BNattention-based NRM, BNattention-based NRM without Dropout の 4 つのモデルを評価する.

### 6.1 データ

本実験では, 2015 年 8 月 19 日から 2015 年 11 月 19 日の 3ヶ月間に収集された英語の会話ツイートを用いる. ツイートの前処理には下記のスクリプト [*4] を修正して用いた. 数字, エクスクラメーションやクエスチョンマークなどの記号の繰り返し, 長音符を用いない長音はそれぞれ<NUMBER>, <REPEAT>, <LONG>のタグトークンに置換される. 修正点は, 1) ユーザ名,URL,emoticon, ハッシュタグのタグトークンの除去, 2) 機種依存文字を含む絵文字及び異体字セレクタの除去, 3)HTML エンティティの除去, 4) 連続するタグの除去である. また, オウム返しを含む会話, 3 人以上のユーザが関連する会話は除去する. 前処理後, spaCy[*5] を用いて各ツイートをトークナイズした. 最頻出の 38398 トークンを学習する語彙として扱う. 未知語は全て<UNK >に置き換えられる.

最大シーケンス長を 32 に設定する. このとき, データ全体の 99.9% をカバーできる. また, シーケンス長の短いツイートが, コンテキストを理解した応答生成の学習を阻害していることが実験的に判明したので, 最短シーケンス長を 12 に設定した. 会話は重複のないように時間軸に沿ってペアに分割した. さらに,25%以上を<UNK >が占める, あるいは 50% 以上をタグトークンが占めるツイートを含むペアも除去した.

我々はデータを訓練, 検証, テストそれぞれ 95, 2.5, 2,5% の割合で分割し, 2743270, 144412, 144110 のツイートペアを得た.

∗4 http://nlp.stanford.edu/projects/glove/preprocess-twitter.rb
∗5 https://spacy.io/

Table 1: 学習結果
Perplexity は低い方が良い性能であることを示す. 訓練時間 H は Early stopping または, 10 エポック終了時の累算である. ただし, no-attention および attention NRM は収束していない.

| NRM | Perplexity | 訓練時間 (H) |
|---|---|---|
| no-attention | 19.24 | 35.5 |
| attention | 18.74 | 52.1 |
| BNattention | 18.37 | 49.7 |
| BNattention without Dropout | **17.26** | **37.1** |

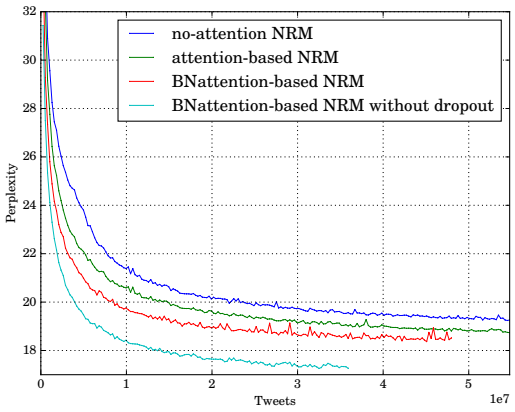


Figure 1: 学習曲線
縦軸は検証データに対する Perplexity
横軸は訓練に用いたツイート数

### 6.2 訓練

単語埋め込み, 隠れ層と readout 層の次元数をそれぞれ, 512, 1024, 1024 とする. 各重み行列は, 標準偏差 0.0001 のガウス分布から初期化される. ただし, 再帰結合の重みは, 直交に初期化する [Saxe 13]. 最適化には, 確率的勾配法の一種である Adam[Kingma 14] をデフォルトパラメータで用いる. また, L2 荷重減衰項の係数 $\lambda$ を 0.00001 とする. [Krizhevsky 12] と同様に, 我々は, この小さな荷重減衰項が, ただの正則化のためだけでなく, DNN において安定した継続的な訓練と誤差削減に寄与することを実験的に発見した. $\mathcal{D}$ を適用するモデルの保持確率は 0.5 とする. また,gradient clipping[Pascanu 12] を閾値 1.0 で適用する. バッチサイズは 128 で固定する. ただし, 各ミニバッチは 4 刻みのシーケンス長でランダムにシャッフルされる. ミニバッチ内の非パディングトークン数を可能な限り減らすことは, RNNs の訓練時間削減に対して非常に有効である.

### 6.3 評価手法

応答生成の自動評価は, 未解決問題のひとつである. 機械翻訳で用いられる BLEU は有効でないことが指摘されている [Ritter 11]. 本実験では, 統計的言語モデルの Perplexity を用いる. ただし, Perplexity は, 生成された応答の自然さを評価するには不十分であることには留意しなければならない. 我々は現在, 単語埋め込み行列と隠れ層のにもとづく, semantic similarity の評価を検討している.

### 6.4 結果

表 1 から, 提案する BNattention-based NRM without Dropout は, attention-based NRM と比べて, 少なくとも 1.4 倍以上高速に, そして, およそ 1.5 低い Perplexity を獲得できたことがわか

Table 2: テストデータに対する応答生成の例

| | |
|---|---|
| Query1 | <NUMBER> th , and yes , i recently had my first trip there <EOS> |
| Response1 | aw <REPEAT> that 's so exciting ! <REPEAT> <EOS> |
| Response2 | i 'm glad you enjoyed it ! <REPEAT> <EOS> |
| Response3 | i 'm glad you had a good trip <EOS> |
| Query2 | i just spent <NUMBER> minutes of my life watching joe sugg play a video game . <REPEAT> and i 'm about to do it again . <EOS> |
| Response1 | i 'm watching it right now . <REPEAT > <EOS > |
| Response2 | do you have a link to the video ? <EOS > |
| Response3 | do you have a link to the game ? <EOS > |
| Response4 | i 'm so glad i 'm not the only one watching this <EOS > |

る. また, 検証データに対する学習曲線を図 1 に示す. さらに, 窓幅 30 のビームサーチによるテストデータに対する応答生成の例を表 2 に示す.

## 7. 結論

我々は, 会話ツイートを大規模収集するシステムを開発し, 収集した会話データに対して, 安定的な最適化と高速な訓練を行える batch normalized attention-based neural responding machine を提案した. 現在, リアルタイムでの学習手法, そして, batch normalized attention mechanism を活かしたより深いネットワーク, より大量の会話ツイートでの学習を検討している.

## References


[Amodei 15] Amodei, D., Anubhai, R., Battenberg, E., Case, C., Casper, J., Catanzaro, B., Chen, J., Chrzanowski, M., Coates, A., Diamos, G., et al.: Deep Speech 2: End-to-End Speech Recognition in English and Mandarin, *arXiv preprint arXiv:1512.02595* (2015)

[Bahdanau 14] Bahdanau, D., Cho, K., and Bengio, Y.: Neural machine translation by jointly learning to align and translate, *arXiv preprint arXiv:1409.0473* (2014)

[Cho 14] Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y.: Learning phrase representations using RNN encoder-decoder for statistical machine translation, *arXiv preprint arXiv:1406.1078* (2014)

[Goodfellow 13] Goodfellow, I. J., Warde-Farley, D., Mirza, M., Courville, A., and Bengio, Y.: Maxout networks, *arXiv preprint arXiv:1302.4389* (2013)

[Hochreiter 97] Hochreiter, S. and Schmidhuber, J.: Long short-term memory, *Neural computation*, Vol. 9, No. 8, pp. 1735–1780 (1997)

[Ioffe 15] Ioffe, S. and Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift, *arXiv preprint arXiv:1502.03167* (2015)

[Kingma 14] Kingma, D. and Ba, J.: Adam: A method for stochastic optimization, *arXiv preprint arXiv:1412.6980* (2014)

[Krizhevsky 12] Krizhevsky, A., Sutskever, I., and Hinton, G. E.: Imagenet classification with deep convolutional neural networks, in *Advances in neural information processing systems*, pp. 1097–1105 (2012)

[Laurent 15] Laurent, C., Pereyra, G., Brakel, P., Zhang, Y., and Bengio, Y.: Batch Normalized Recurrent Neural Networks, *arXiv preprint arXiv:1510.01378* (2015)

[Morstatter 13] Morstatter, F., Pfeffer, J., Liu, H., and Carley, K. M.: Is the sample good enough? comparing data from twitter's streaming api with twitter's firehose, *arXiv preprint arXiv:1306.5204* (2013)

[Pascanu 12] Pascanu, R., Mikolov, T., and Bengio, Y.: On the difficulty of training recurrent neural networks, *arXiv preprint arXiv:1211.5063* (2012)

[Ritter 11] Ritter, A., Cherry, C., and Dolan, W. B.: Data-driven response generation in social media, in *Proceedings of the conference on empirical methods in natural language processing*, pp. 583–593Association for Computational Linguistics (2011)

[Saxe 13] Saxe, A. M., McClelland, J. L., and Ganguli, S.: Exact solutions to the nonlinear dynamics of learning in deep linear neural networks, *arXiv preprint arXiv:1312.6120* (2013)

[Schuster 97] Schuster, M. and Paliwal, K. K.: Bidirectional recurrent neural networks, *Signal Processing, IEEE Transactions on*, Vol. 45, No. 11, pp. 2673–2681 (1997)

[Shang 15] Shang, L., Lu, Z., and Li, H.: Neural responding machine for short-text conversation, *arXiv preprint arXiv:1503.02364* (2015)

[Sordoni 15] Sordoni, A., Galley, M., Auli, M., Brockett, C., Ji, Y., Mitchell, M., Nie, J.-Y., Gao, J., and Dolan, B.: A neural network approach to context-sensitive generation of conversational responses, *arXiv preprint arXiv:1506.06714* (2015)

[Srivastava 14] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R.: Dropout: A simple way to prevent neural networks from overfitting, *The Journal of Machine Learning Research*, Vol. 15, No. 1, pp. 1929–1958 (2014)

[Wen 15] Wen, T.-H., Gasic, M., Kim, D., Mrksic, N., Su, P.-H., Vandyke, D., and Young, S.: Stochastic language generation in dialogue using recurrent neural networks with convolutional sentence reranking, *arXiv preprint arXiv:1508.01755* (2015)

[Xu 15] Xu, K., Ba, J., Kiros, R., Courville, A., Salakhutdinov, R., Zemel, R., and Bengio, Y.: Show, attend and tell: Neural image caption generation with visual attention, *arXiv preprint arXiv:1502.03044* (2015)