

# LSTMを用いたパーソナル対話技術

## Personalized chatbot based on LSTM

奥井 颯平

Sohei Okui

中辻 真

Makoto Nakatsuji

NTT レゾナント

NTT Resonant, Inc

LSTMを用いたチャットボットシステムへの研究開発が注目されている。しかし、現状のチャットボットは全体最適化された AI 応答を返すものが主であり、個人の特性に応じた AI 応答は返却できていない。この問題を解決するため、本稿では、以下 2 点の提案を行う。(1) 現在のユーザ発話に応じた個人化された AI 応答を行うため、ユーザの過去の発話列から、現在の発話と関係付けられるトピックをアテンションモデルを応用して抽出し、ユーザの現在の発話の表現ベクトルに反映する。それにより、ユーザの現在の発話と、過去の発話履歴、両方に応じた、個人化された AI 応答選択が実現できる。また、(2) 選択された AI 応答のユーザ発話とのマッチングやユーザの反応を基に AI 応答をオンライン学習する枠組みを提案し、より個人化された AI 応答選択を目指す。

### 1. はじめに

LSTMを用いた対話応答学習とコミュニケーションサービスへの応用が注目されている。例えば、ドラマのチャットボット「AI カホコ」\*1、女子高生 AI チャットボット「AI りんな」、恋愛相談 AI 「AI オシエル」\*2 などが有名である。これらのサービスでは、インスタントメッセージサービスや Q&A コミュニティサービス上に AI キャラクタが登場し、ユーザと、雑談や恋愛相談といったコミュニケーションを行う。特に、AI カホコ等のチャットボットのように、LINE 等のインスタントメッセージサービス上に登場した AI キャラクタとユーザは、その場限りではなく、日をまたがって継続的にコミュニケーションを楽しむことが多くなっている。そうした中では、各ユーザに対して、個人化された AI 応答を提供することが、今後のユーザ満足度向上に繋がる可能性があると考えられる。しかし、現状のチャットボットは、ユーザの名前や性別などの登録されたユーザ情報や、GPS から把握される生活エリアに基づくルールベースでの個人化しか実現できていない。つまり、ユーザの現在の発話を基にして次の AI 応答を選択しているのみであり、ユーザと AI キャラクタとの過去のコミュニケーションを活かしきれていない。そのため、実際に人間の間で、コミュニケーションを円滑にしたり、親しみを感じさせるために重要になってくる、過去のやり取りから推定されるユーザ興味や性格に応じた気の利いた AI 応答は実現できていない。

本稿では、上述した課題を解決するため、我々が過去に提供してきた「AI オシエル」や「AI カホコ」で用いてきた LSTM モデルを拡張し、個人化された AI 応答を実現する Personalized LSTM モデル (PLSTM) を提案する。PLSTM におけるアイデアは以下の 2 つである。(1) ユーザと AI の過去のコミュニケーション履歴を現在のユーザ発話への AI 応答に用いる。特に、時時刻刻と変化するユーザ興味や状況に対応するため、PLSTM は、Encoder-Decoder モデルで利用されている Attention メカニズムを応用し、ユーザの過去の発話列の中で、現在のユーザ発話と関連性の高いトピックを抽出し、現在のユーザ発話へ反映させる。このように過去の関連するトピックを現発話に反映させることで、より個人化されたユーザ満足度の高い AI 応答

選択に繋がると考えられる。(2) 選択された AI 応答へのユーザの反応を基に AI 応答の適切さを学習する。特に本稿では AI 応答の適切さを以下の 3 つの尺度、(a) ユーザ発話と AI 応答のマッチング、(b) AI 応答に対するユーザ応答の早さ、(c) AI 応答へのユーザ応答のポジティブさ、を基にして取り扱う。これは過去の弊社の各種チャットボットの運用により得られた知見として、継続的に AI とのコミュニケーションを楽しむユーザは、ユーザと AI の会話が成立しており、AI 応答へのユーザ応答も早く、応答の中に満足度の高い表現 (例えば、「いいね!」や「面白いね」など) が多く、満足度の低い表現 (例えば、「つまらない」など) が少ない傾向があることに基づく。そのため、学習関数の中に、上記 3 点の尺度を反映し、よりユーザ満足度の高い個人化された AI 応答を目指す。

### 2. 関連研究

深層学習を用い、対話を個人化する試みは、深層学習を用いた情報推薦の研究 [Elkahky 15] に近い。深層学習ではない試みとしては、明示的な概念体系であるセマンティクスを用いて、直近のユーザ行動と、興味概念が体系的に近い概念に属する過去のユーザ行動を組み合わせ、次のユーザへの推薦を決定する研究がある [Nakatsuji 12]。しかし、本研究のように直近のユーザ発話を、潜在的に関連する過去のユーザ発話トピックを用い、アテンションメカニズムにより強化する研究ではない。[Elkahky 15] は深層学習を用いて、ユーザ興味を抽出し、複数ドメインをまたがる推薦を実現する。しかし、ユーザの直近の興味に焦点をあてる研究ではない。

一方、近年の深層強化学習に基づくモデルでは、2 つの仮想エージェント間の対話をシミュレートし、ポリシー勾配法を使用して、3 つの有用な対話プロパティ、(1) 対話の持つ情報量、(2) 一貫性、および (3) 応答の容易さを持つ対話シーケンスに報酬を与えている [Li 16]。その結果、彼らの実験では対話の継続性や多様性が向上したことが報告されている。また、クラウドソーシングされたデータと現実のユーザとのやりとりを強化学習を適用することで、適切な AI 応答を選択するように制御する試みもあり、実験により高精度な結果が確認されている [Serban 18]。こうした強化学習によるチャットボットの研究は、チャットボットの個人化の学習にも応用できると考えられる。なお、上述した 2 つの強化学習の研究のように、AI 応答

連絡先: 中辻 真, nakatsuji@nttr.co.jp

\*1 <http://www.ntv.co.jp/kahogo-kahoko/special/index.html>

\*2 <https://oshiete.goo.ne.jp/ai>

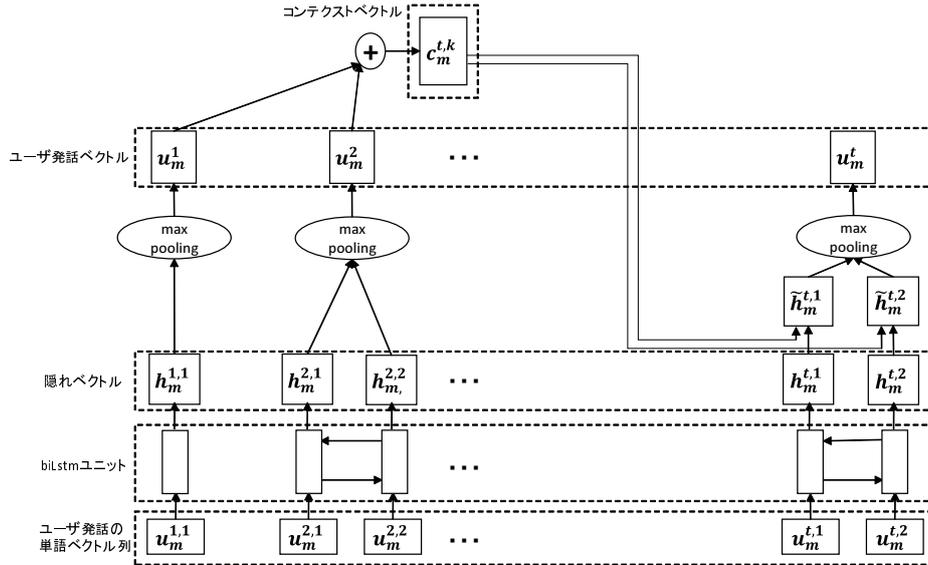


図 1: PLSTM におけるアテンションメカニズムのイメージ図

には、生成と選択のアプローチがあるが、本稿では、著者らの一次検証において、応答生成は応答選択タスクに比べて性能が劣る傾向が見られた。そのため、本稿では、応答選択タスクである QA-LSTM フレームワーク上に提案手法を展開していく。

### 3. QA-LSTM

本章では、提案手法の土台となる QA-LSTM フレームワークについて説明する。

はじめに、LSTM について説明するとともに、本稿における用語を定義する。入力文を  $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}$  とする。ここで、 $\mathbf{x}_t$  は  $t$  番目の単語ベクトルである。 $t$  番目の隠れベクトル  $\mathbf{h}_t$  は以下の式で更新される。

$$\begin{aligned} \mathbf{i}_t &= \sigma(\mathbf{W}_i \mathbf{x}_t + \mathbf{U}_i \mathbf{h}_{t-1} + \mathbf{b}_i) \\ \mathbf{f}_t &= \sigma(\mathbf{W}_f \mathbf{x}_t + \mathbf{U}_f \mathbf{h}_{t-1} + \mathbf{b}_f) \\ \mathbf{o}_t &= \sigma(\mathbf{W}_o \mathbf{x}_t + \mathbf{U}_o \mathbf{h}_{t-1} + \mathbf{b}_o) \\ \tilde{\mathbf{c}}_t &= \tanh(\mathbf{W}_c \mathbf{x}_t + \mathbf{U}_c \mathbf{h}_{t-1} + \mathbf{b}_c) \\ \mathbf{c}_t &= \mathbf{i}_t * \tilde{\mathbf{c}}_t + \mathbf{f}_t * \mathbf{c}_{t-1} \\ \mathbf{h}_t &= \mathbf{o}_t * \tanh(\mathbf{c}_t) \end{aligned}$$

LSTM ブロックは 3 つのゲート (入力ゲート  $\mathbf{i}_t$ 、忘却ゲート  $\mathbf{f}_t$ 、出力ゲート  $\mathbf{o}_t$ ) とメモリセル  $\mathbf{c}_t$  から構成される。 $\sigma$  はシグモイド関数である。 $\mathbf{W} \in R^{H \times N}$ 、 $\mathbf{U} \in R^{H \times H}$ 、 $\mathbf{b} \in R^{H \times 1}$  は本ネットワークにおける学習パラメータである。単方向の LSTM の場合、各トークンより以前の情報を引き継いで文脈情報を学習していく一方で、各トークンの後に現れるトークンからの文脈情報を考慮することが難しい。BiLSTM は、前後の文脈情報を双方向に処理し、2 つの出力ベクトルを得る。BiLSTM ブロックの出力は、前後方向の出力 2 つを以下の式のように連結したものである  $\overrightarrow{h}(t) = \overrightarrow{h}(t) \parallel \overleftarrow{h}(t)$ 。

QA-LSTM[Tan 16] のフレームワークは以下の通りである。質問  $q$ 、回答候補  $a$  からなる、与えられた入力ペア  $(q, a)$  に対し、 $q$ 、 $a$  それぞれについて単語ベクトル化する。次に、それぞれ別々の BiLSTM ブロックに単語ベクトル化したシーク

エンスを入力として与える。BiLSTM から出力されたベクトルに対して max pooling を行い、双方向分の 2 つの出力ベクトルを連結することにより、 $q$  においては  $\mathbf{o}_q$ 、また  $a$  においては  $\mathbf{o}_a$  として固定長の分散表現にする。最後に、入力ペア  $(q, a)$  に対してコサイン類似度  $\cos(\mathbf{o}_q, \mathbf{o}_a)$  によってスコアリングを行う。

学習要素は以下のヒンジロス関数によって定義する。

$$\mathcal{L} = \max\{0, M - \cos(\mathbf{o}_q, \mathbf{o}_a^+) + \cos(\mathbf{o}_q, \mathbf{o}_a^-)\} \quad (1)$$

ここで  $\mathbf{o}_a^+$  は本来の正解の回答の出力ベクトルであり、 $\mathbf{o}_a^-$  は全回答空間からランダムに選出された不正解の回答の出力ベクトルである。また、 $M$  は損失関数におけるマージンである。複数の正解回答を持つ質問については、その質問に紐づく各回答についてそれぞれ複数の別のペアとして扱う。

### 4. 提案手法

本章では提案手法 PLSTM について説明する。まず、アテンションメカニズムによる、ユーザーの過去の発話列を現在の発話と関係付ける方法について述べ、その次に、ユーザー毎に学習された特性に基づく応答の個人化について述べる。

#### 4.1 アテンションメカニズム

本稿で提案するアテンションメカニズムについて説明する。図 1 に提案するアテンションメカニズムのイメージを示す。

以下に用語を定義する。ユーザーの数を  $M$  とし、 $m$  番目のユーザーの発話を  $\mathbb{U}_m = \{\mathbf{u}_m^1, \mathbf{u}_m^2, \dots, \mathbf{u}_m^t\}$  とする。ここで、 $\mathbf{u}_m^t$  は  $t$  番目の発話ベクトルである。各発話は、複数の単語トークンからなる。例えば  $t$  番目の発話列は  $\{\mathbf{u}_m^{t,1}, \mathbf{u}_m^{t,2}, \dots, \mathbf{u}_m^{t,n_m^t}\}$  と表される。ここで、 $n_m^t$  は  $m$  番目のユーザーの  $t$  番目のユーザー発話に含まれる単語長であり、 $\mathbf{u}_m^{t,k}$  は  $m$  番目のユーザーの  $t$  番目の発話の中の  $k$  番目の単語ベクトルである。なお、ユーザー発話  $\mathbf{u}_m^t$  に対応する応答は、 $\mathbf{a}_m^t$  であり、その発話ベクトルは  $\mathbf{a}_m^t$ 、発話列は  $\{\mathbf{a}_m^{t,1}, \mathbf{a}_m^{t,2}, \dots, \mathbf{a}_m^{t,n_m^t,a}\}$  と表される。ここで、 $n_m^t,a$  は  $m$  番目のユーザーの  $t$  番目のユーザー発話への AI 応答に含まれる単語長であり、 $\mathbf{a}_m^{t,k}$  は  $m$  番目のユーザーの  $t$  番目の発話への AI 応答の中の  $k$  番目の単語ベクトルである。

次にアテンションを現在の発話ベクトル  $\mathbf{u}_m^t$  に反映する手順について述べる。

(1) データセットからあるユーザ  $m$  の現在の発話  $u_m^t$  を一つピックアップする。

(2) ユーザ  $m$  の過去の発話系列  $\{u_m^1, u_m^2, \dots, u_m^{t-1}\}$  の各発話に対し、QA-LSTM と同様の方法で発話ベクトルを計算する。例えば  $t-1$  番目の発話ベクトルは、QA-LSTM と同様に、その発話内の単語列に BiLSTM を適用し、出力される  $n_m^{t-1}$  個の隠れベクトル  $\{\mathbf{h}_m^{t-1,1}, \dots, \mathbf{h}_m^{t-1,n_m^{t-1}}\}$  に対し max-pooling を適用することで、発話ベクトル  $\mathbf{u}_m^{t-1}$  を獲得する。

(3) そして、 $t$  番目の発話ベクトル  $\mathbf{u}_m^t$  を計算する際に、(2) で計算した過去の発話ベクトルを  $t$  番目の現在の発話の  $k$  番目の単語へどれだけ反映するかを決定するコンテキストベクトル  $\mathbf{c}_m^{t,k}$  を以下のように計算する。なお、 $\mathbf{V}_a$ 、 $\mathbf{W}_a$ 、 $\mathbf{U}_a$  は重み行列である。

$$\mathbf{c}_m^{t,k} = \sum_{j=1}^{t-1} \alpha_{k,j} \mathbf{u}_m^j,$$

$$\alpha_{k,j} = \frac{\exp(e_{k,j})}{\sum_{l=1}^{t-1} \exp(e_{k,l})},$$

$$e_{k,j} = \mathbf{V}_a^T \tanh(\mathbf{W}_a \mathbf{h}_m^{t,k} + \mathbf{U}_a \mathbf{u}_m^j)$$

このコンテキストベクトルの計算方法は、[Bahdanau 14] における方法を基にしているが、[Bahdanau 14] は Encoder-Decoder model にアテンションを適用している。一方、提案手法は、同一の発話ストリームの中で、現在の発話に過去の発話からアテンションを計算している点で大きく異なる。さらに、提案手法では、過去のユーザの興味は抽象的に捉えることが重要であると考えている。そのため、複数の単語ストリームからより抽象度の高いユーザ発話ベクトルを計算し、各ユーザ発話ベクトルをどの程度、現在のユーザ発話を構成する単語ストリーム内の隠れベクトルに反映するかを決定している。これは、[Bahdanau 14] のように、一連の単語ストリームの中のみでアテンションを計算している従来方式とは異なる。

次にコンテキストベクトル  $\mathbf{c}_m^{t,k}$  を BiLSTM に以下のように反映させることで、提案手法は、過去のユーザ発話を現在の発話の  $k$  番目の隠れベクトル  $\mathbf{h}_m^{t,k}$  へ反映することができるようになる。

$$\mathbf{z}_k = \sigma(\mathbf{W}_z[\mathbf{u}_m^{t,k-1}, \mathbf{T}] + \mathbf{U}_z \mathbf{h}_m^{t,k-1} + \mathbf{W}_z^a \mathbf{c}_m^{t,k} + \mathbf{b}_z)$$

$$\tilde{\mathbf{l}}_k = \tanh(\mathbf{W}_l[\mathbf{u}_m^{t,k-1}, \mathbf{T}] + \mathbf{U}_l \mathbf{h}_m^{t,k-1} + \mathbf{W}_l^a \mathbf{c}_m^{t,k} + \mathbf{b}_l)$$

$$\mathbf{l}_k = \mathbf{i}_k * \tilde{\mathbf{l}}_k + \mathbf{f}_k * \mathbf{l}_{k-1}$$

$$\mathbf{h}_m^{t,k} = \mathbf{o}_k * \tanh(\mathbf{l}_k)$$

ここで、 $z$  は入力  $i$ 、忘却  $f$ 、出力  $o$  を示すトークンであり、入力ゲート  $\mathbf{i}_k$ 、忘却ゲート  $\mathbf{f}_k$ 、出力ゲート  $\mathbf{o}_k$  に対応する。 $\mathbf{l}_k$  はセルメモリベクトルである。 $\mathbf{W}_z^a$  と  $\mathbf{W}_l^a$  はアテンションパラメータである。

(4)  $t$  番目の発話ベクトル  $\mathbf{u}_m^t$  は、上記手順で更新された  $n_m^t$  個の隠れベクトル  $\{\tilde{\mathbf{h}}_m^{t,1}, \dots, \tilde{\mathbf{h}}_m^{t,n_m^t}\}$  に対し max-pooling を適用することで獲得できる。

#### 4.1.1 AI 応答の個人化の例

発話応答の個人化の例を以下に示す。

例えば AI キャラクタに対するユーザ発話に「大好きだよ」という履歴があり、現在の発話が「久しぶりだね」である場合、

ユーザは AI キャラクタと親密なやり取りを行う傾向があり、結果として「会いたかった」と親密さを演出する応答を行う。一方、ユーザ発話に「眠い今日も疲れた」という履歴があり、現在の発話が「久しぶりだね」である場合、ユーザは日常の愚痴をチャットボットにこぼす傾向があり、ユーザを心配する「元気にしてた？」を応答できる。このように同じ「久しぶりだね」という現在の発話に対し、異なる応答を個人毎に返す。

## 4.2 ユーザ毎の応答への特性の学習

次に、ユーザごとの応答への特性の学習について述べる。提案手法では、(1) ユーザ発話と応答とのマッチング、(2) ユーザの応答へのレスポンス、(3) ユーザの応答への満足度、の3点を基にユーザ発話と応答のやりとりを学習する。

### 4.2.1 ユーザ発話と応答とのマッチング

ユーザ発話と応答とのマッチングについては、QA-LSTM と同様にして、発話ベクトルと応答ベクトルの類似性に基づき以下の式で計算する。

$$r_1 = \cos(\mathbf{u}_t, \mathbf{a}_t^+) - \cos(\mathbf{u}_t, \mathbf{a}_t^-) \quad (2)$$

ここで  $\mathbf{a}_t^+$  は観測時に収集された本来の正解の応答の出力ベクトルであり、 $\mathbf{a}_t^-$  は全応答空間からランダムに選出された不正解の応答の出力ベクトルである。

### 4.2.2 ユーザの応答へのレスポンス速度

AI 応答  $a_t$  の会話の盛り上がりへの貢献については、 $a_t$  へのユーザ返答の素早さが尺度となると考えられる。評価する AI 応答を  $a_t$  とし、 $a_t$  の次のユーザ発話を  $u_{t+1}$  とする。そして、ユーザが  $u_{t+1}$  を返答するまでにかかる時間を  $s_t$  とする。このとき、ユーザの応答へのレスポンス速度に基づく AI 応答  $a_t$  への貢献度は、以下の式で計算する。

$$r_2 = \frac{1}{s_t} \quad (3)$$

### 4.2.3 ユーザの応答への満足度

AI 応答  $a_t$  のユーザ満足度への貢献については、[Serban 18] と同様に、 $a_t$  へのユーザの返答  $u_{t+1}$  に、“いいね”などの高い満足度を示す単語が含まれるか、もしくは、“意味不明”などの低い満足度を示す単語が含まれるかを基に計測できる。そこで、高い満足度を示す単語の集合を  $\mathbb{P}$  とし、低い満足度を示す単語の集合を  $\mathbb{N}$  とする。このとき、ユーザ満足度に基づく AI 応答  $a_t$  への貢献度は、以下の式で計算される。

$$\begin{aligned} r_3 &= 3 \quad \text{s.t.} \quad u_{t+1} \in \mathbb{P} \\ &= 2 \quad \text{s.t.} \quad u_{t+1} \notin \{\mathbb{P} \cup \mathbb{N}\} \\ &= 1 \quad \text{s.t.} \quad u_{t+1} \in \mathbb{N} \end{aligned}$$

### 4.2.4 損失

最終的な損失  $\mathcal{L}$  は、QA-LSTM の損失の設計にふまえて、ユーザ毎の応答への特性を考慮し、以下のように設計する。

$$\mathcal{L} = \max\{0, M - \mathcal{N}(r_1)\mathcal{N}(r_2)\mathcal{N}(r_3)\} \quad (4)$$

ここで、 $\mathcal{N}$  は、ユーザ発話と応答とのマッチングや、ユーザの応答へのレスポンス速度や、ユーザの応答への満足度のスコアを 0 から 1 の範囲の一様分布に変換する関数である。また、 $M$  はマージンである。上記損失は、マッチングが高く、レスポンス速度や、応答への満足度が高いほど、損失が低くなるように設計されている。そして、本損失に基づき、LSTM に関するパラメータ群は、確率的勾配降下法と誤差逆伝播法を用いて学習される。

## 5. 結論

本稿では、現在の発話に基づく AI 応答を、個人毎に特化したものとするため、(1) ユーザの過去の発話列から重要なトピックを現在のユーザ発話へ反映し、次の AI 応答の選択に活用すること、(2) AI 応答へのユーザの返答の素早さ、満足度を基にユーザ毎の会話を学習すること、を提案した。現在、提案手法を実際の会話ログに適用することで、モデルの有効性の検証を進めている。さらに、ユーザ毎の応答への特性を学習するにあたり、本稿のように損失関数に上記特性を反映させる方式と、[Li 16] の研究のように、強化学習にもとづく方式との比較検討を進める。

## 参考文献

- [Bahdanau 14] Bahdanau, D., Cho, K., and Bengio, Y.: Neural Machine Translation by Jointly Learning to Align and Translate, *CoRR*, Vol. abs/1409.0473, (2014)
- [Elkahky 15] Elkahky, A. M., Song, Y., and He, X.: A Multi-View Deep Learning Approach for Cross Domain User Modeling in Recommendation Systems, in *Proc. WWW'15*, pp. 278–288 (2015)
- [Li 16] Li, J., Monroe, W., Ritter, A., Jurafsky, D., Galley, M., and Gao, J.: Deep Reinforcement Learning for Dialogue Generation., in *Proc. EMNLP*, pp. 1192–1202 (2016)
- [Nakatsuji 12] Nakatsuji, M., Fujiwara, Y., Uchiyama, T., and Toda, H.: Collaborative Filtering by Analyzing Dynamic User Interests Modeled by Taxonomy., in *Proc. ISWC'12*, pp. 361–377 (2012)
- [Serban 18] Serban, I. V., Sankar, C., Germain, M., Zhang, S., Lin, Z., Subramanian, S., Kim, T., Pieper, M., Chandar, S., Ke, N. R., Rajeswar, S., Br bisson, de A., Sotelo, J. M. R., Suhubdy, D., Michalski, V., Nguyen, A., Pineau, J., and Bengio, Y.: A Deep Reinforcement Learning Chatbot (Short Version), *CoRR*, Vol. abs/1801.06700, (2018)
- [Tan 16] Tan, M., Santos, dos C. N., Xiang, B., and Zhou, B.: Improved Representation Learning for Question Answer Matching, in *Proc. ACL'16*, pp. 464–473 (2016)