The 33rd Annual Conference of the Japanese Society for Artificial Intelligence, 2019

言語モデルによる文の最適分割に基づく音声言語理解

Spoken Language Understanding based on Sentence Segmentation by Language Models

若林 啓 *1 Kei Wakabayashi　竹内 誉羽 *2 Johane Takeuchi　平松 淳 *1 Makoto Hiramatsu　中野 幹生 *2 Mikio Nakano

*1筑波大学 University of Tsukuba　*2（株）ホンダ・リサーチ・インスティチュート・ジャパン Honda Research Institute Japan Co., Ltd.

In this paper, we propose a new approach to solve the slot filling task for spoken language understanding by using a formulation based on the optimum segmentation of an input sentence. This formulation enables us to develop a language modeling-based method that is drastically efficient compared to the existing deep learning approach that formalizes the slot filling as a sequence labeling task. The proposed method trains the language models by a one-pass algorithm and applies a dynamic programming algorithm to find the most likely slot assignment efficiently. We empirically confirmed that the proposed method achieves a competitive accuracy compared to a deep learning method, and even works with drastically less computing resource consumption.

1. はじめに

近年，音声認識システムやチャットボットの実用化が注目を集めている．これらのシステムのインタフェースでは，ユーザは自然言語の文章でシステムに対して指示を行うため，入力された文から機能の実行に必要な情報を抽出する必要がある．例えば，「新潟の降水量を教えて」というユーザの入力に対して，{ **Where**：新潟, **What**：降水量 } といった情報を抽出する．このようなタスクはスロットフィリングと呼ばれ，**Where** や **What** などの変数をスロットと呼ぶ．

スロットフィリングタスクは，系列ラベリングの問題として定式化されることが多く，深層学習を用いた手法によって高い精度で抽出できることが報告されている [Xu 13]．これらの手法では，CNN や LSTM の層によって文章の特徴量を抽出し，タグの前後関係を考慮した CRF などの層によって系列ラベルを予測する [Mesnil 15]．このアプローチは精度が高いが，モデルの学習に大きな計算資源が必要になる．

本研究では，計算資源の小さい環境でも高速に学習が可能なスロットフィリング手法を提案する．提案手法は，スロットごとに学習した言語モデルを用いて，入力文の尤度が最大になるような分割を求めることで，スロットフィリングを実現する．特に，図 1 に示すように，スロット部分の確率だけではなく，非スロット部分も言語モデルを定義して尤もらしさを評価することにより，頑強で精度の高いスロット推定が可能になることを明らかにする．深層学習よりも学習や推定にかかる時間が少ない言語モデルを用いることで，小さい計算資源で深層学習手法と同等の精度が達成できることを実験により示す．

2. 提案手法

本研究では，文のラベル付き分割を行うことでスロットフィリングを実現する．例えば，図 1 の (b) の例では，入力文を「**新潟市**」「**の**」「**降水量**」「**を教えて**」の 4 つのセグメントに分割し，それぞれ「**Where スロット**」「**Middle 非スロット**」「**What スロット**」「**Ending 非スロット**」というラベルを付与している．形式的には，入力文をトークンの系列 $x_{1:T} = x_1, \ldots x_T$ と表し *1，各セグメントの最後のトークンの位置を $b_1, \ldots, b_K$，それぞれのセグメントに付与するラベルを $y_1, \ldots, y_K$ と表す．k 番目のセグメントに対応する部分トークン列は $s_k = x_{b_{k-1}+1:b_k}$ と表すことができる．ただし，最後の分割位置は必ず系列の終端であるため $b_K = T$ であり，便宜的に $b_0 = 0$ とする．

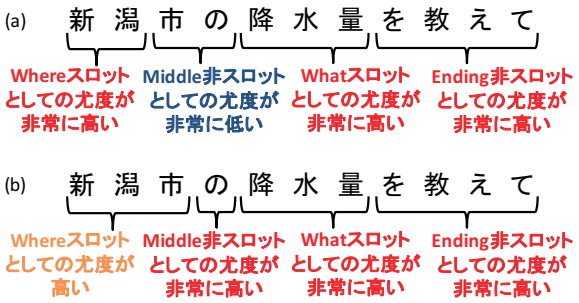

図 1: 文の最適分割に基づくスロットフィリング．提案手法は非スロット部分も含めた結合尤度をモデル化するため，(a) よりも (b) の分割を最適分割として選択する．

スロットフィリングタスクでは，抽出を要求されるスロットの集合 $\mathcal{Z}$（例えば，{ **Where**, **When**, **What** }）と，学習データの集合が与えられる．学習データは，文とスロット値割り当てのペアであり，例えば図 1 の文に対して { **Where**：**新潟**, **What**：**降水量** } の情報が付与されているものとする．本研究では，この学習データをラベル付き分割の形式に変換する．この変換は，以下のステップによって行われる．

- 学習データに付与されているそれぞれのスロット $z \in \mathcal{Z}$ に対応する値を，トークン列の完全マッチに基づいて文の中から見つける．文におけるマッチした箇所をセグメントとし，ラベル z を割り当てる．
- 全てのスロットについて上記の処理を行なった後に，マッチしなかった箇所のトークン列を，それぞれセグメントとする．これらのセグメントに対応するラベルは，もしセ

連絡先: 若林啓，筑波大学図書館情報メディア系，茨城県つくば市春日 1-2，kwakaba@slis.tsukuba.ac.jp

*1　本論文の実験では，日本語の場合は文字を，英語の場合は単語を，それぞれトークンとして用いる．

グメントが文の先頭にあれば「**Beginning 非スロット**」，文の末尾にあれば「**Ending 非スロット**」，それ以外ならば「**Middle 非スロット**」とする．

このため，上記の 3 種類の非スロットラベルの集合を $\mathcal{U}$ とすると，ラベルの集合は $\mathcal{Y} = \mathcal{Z} \cup \mathcal{U}$ となる．

本研究では，推論アルゴリズムの計算量を抑えるため，セグメントの前後関係については考慮しない．入力トークン列とラベル付き分割の同時確率を，以下のように定義する．

$$p(x_{1:T}, b_{1:K}, y_{1:K}) = \prod_{k=1}^{K} p(y_k) p(x_{b_{k-1}+1:b_k} | y_k) \quad (1)$$

$p(y_k)$ はラベルの出現確率であり，学習データ中の頻度に基づいて推定する．$p(x_{b_{k-1}+1:b_k}|y_k)$ は，ラベル y_k に対応した言語モデルによるトークン列の確率であり，以下の節で述べる．

2.1 言語モデルの定義

それぞれのラベル $y \in \mathcal{Y}$ について，異なるパラメータを持つ言語モデル $\mathcal{P}_y$ を仮定する．ここでは，言語モデルは，トークン列 $s = w_1, \ldots, w_L$ に対して確率を割り当てる確率モデルであり，その標本空間 $\mathcal{V}$ は以下で表される．

$$\mathcal{V} = \{w_1, \ldots, w_L | w_t \in \mathcal{C}, L \leq 0\}$$

$\mathcal{V}$ の要素であるトークン列を，フレーズと呼ぶ．スロットフィリングタスクにおいては，実際にスロットに割り当てられるフレーズの種類は少ない場合が多い．例えば，前述の例において，「**Where スロット**」には，ほとんどの場合は都道府県や市区町村の名前が指定されるため，同じフレーズが繰り返し出現することが予想される．この仮定に基づいて，本研究では，中華料理店過程（CRP）を言語モデルとして用いる．CRP は，標本空間 $\mathcal{V}$ 上の確率分布であり，特に過去に観測したことのある値に高い確率を割り当てる．CRP は，観測したことのない値に対しても，基底分布と呼ばれる別の確率分布に基づいて確率を割り当てることから，事前にスロットが取り得る値の集合を決定することなく，特定の種類のフレーズに集中して確率を割り当てた分布を推定することができる．本研究では，CRP の基底分布として n-gram モデルを用いる．

2.1.1 n-gram モデル

n-gram モデルは，過去の n トークンに依存した離散分布に基づいて次のトークンを推定する確率モデルである．ここでは，$\mathcal{V}$ 上での確率の総和が 1 になる確率分布を定義するために，フレーズの長さの確率を明示的にモデル化する [Zhai 13]．フレーズ $s = w_1, \ldots, w_L$ の確率は，長さが L である確率と，それぞれのトークンの n-gram 確率の積によって定義される．

$$p_{ngram}(w_1, \ldots, w_L) = p(L) \prod_{t=1}^{L} p(w_t | w_{t-n+1:t-1})$$

$p(L)$ は，最大トークン列長を仮定した上で，離散分布によって定義する．実験では，最大トークン列長は 256 とした．

2.1.2 ディリクレ過程スロットモデル

ディリクレ過程 $DP(\alpha^0, G^0)$ は，標本空間 $\mathcal{V}$ 上の離散確率分布を生成する確率分布であり，基底分布と呼ばれる $\mathcal{V}$ 上の確率分布 G^0 と，生成される分布の偏りの大きさを調整するパラメータ α^0 によって定められる．ここでは，$G^0 = p_{ngram}$ とする．フレーズがある離散確率分布 $\mathcal{P}$ に従うとして，$\mathcal{P}$ の事前分布をディリクレ過程 $DP(\alpha^0, p_{ngram})$ とすると，N 個のフレーズの観測値 $s_{1:N} = s_1, \ldots, s_N$ の次に観測されるフレーズ s_{N+1} の予測分布は，CRP と呼ばれる確率過程に従う [Teh 05]．CRP では，補助変数として，基底分布から生成される値を保持するための潜在変数 $\boldsymbol{\phi} = \{\phi_1, \phi_2, \ldots\}$ と，それぞれの観測値が $\boldsymbol{\phi}$ のどの要素に対応するかを表す潜在変数 $c_{1:N} = c_1, \ldots, c_N$ を考える．c_i は，$s_i = \phi_{c_i}$ を満たすような値しか取らないものとする．CRP による予測分布は以下で与えられる．

$$p_{crp}(s_{N+1} | c_{1:N}, \boldsymbol{\phi}, \alpha^0, p_{ngram}) = \sum_{m=1}^{M} \frac{n_m}{N + \alpha^0} \delta(\phi_m, s_{N+1}) + \frac{\alpha^0}{N + \alpha^0} p_{ngram}(s_{N+1})$$

ただし，n_m は $c_{1:N}$ の中で ϕ_m が出現する回数，M は $c_{1:N}$ の中に含まれる値の種類の数，$\delta(\phi_m, s_{N+1})$ は $\phi_m = s_{N+1}$ ならば 1，そうでなければ 0 になる指示関数である．この予測分布は，確率 $\frac{N}{N+\alpha^0}$ で過去の観測値 $s_{1:N}$ の中に出現したフレーズが現れ，確率 $\frac{\alpha^0}{N+\alpha^0}$ で n-gram 分布に従って新たに生成されるフレーズが現れると推定しており，過去に観測した値に高い確率を与える分布になっている．このモデルをディリクレ過程スロットモデル（Dirichlet Process Slot Model, DPSM）と呼ぶ．本研究では，ラベル $y \in \mathcal{Y}$ に対応する言語モデル $\mathcal{P}_y$ はそれぞれ独立なパラメータと潜在変数をもつ DPSM とし，フレーズ $s = w_1, \ldots, w_L$ の確率は p_{crp} によって計算する．

2.2 言語モデルの学習

本研究では，各ラベルの言語モデルは独立に学習を行う．学習データはラベル付き分割の形式であるが，セグメントごとに独立にフレーズを抽出して言語モデルの学習データとする．

提案手法は，学習データ全体を 1 回だけスキャンするワンパスアルゴリズムによってモデルパラメータを学習する．DPSM には，潜在変数として $c_{1:N}$ と $\boldsymbol{\phi}$ が含まれているが，これらは事後確率分布に従ってサンプリングを行うことで，逐次的に確定させる．N 番目の学習データ s_N が与えられたとき，対応する割り当て c_N の事後確率は，以下で与えられる．

$$p(c_N = m | s_N, c_{1:N-1}, \boldsymbol{\phi}) = \begin{cases} \frac{n_m \delta(\phi_m, s_N)}{\sum_{m'} \delta(\phi_{m'}, s_N) + \alpha^0} & m \leq M \\ \frac{\alpha^0}{\sum_{m'} \delta(\phi_{m'}, s_N) + \alpha^0} & m = M+1 \end{cases}$$

もし N 番目までに s_N と同じフレーズが一度も出現しなければ，$m = M+1$ となる確率が 1 になる．

この分布に従ってサンプリングを行い，もし $m = M+1$ が選ばれたならば，遅延評価により $\phi_{M+1} = s_N$ と確定する．このとき，s_N は基底分布である n-gram モデルの $M+1$ 番目の観測値となるため，n-gram モデルのパラメータを更新する．n-gram モデルは，トークンの n-gram 隣接確率と，フレーズ長に関するパラメータを持つが，これらはそれぞれ n-gram 頻度とフレーズ長の数え上げを行うことで更新する．

2.3 動的計画法による最適分割発見

学習済みの言語モデルを用いて，与えられた入力文に対する最適なラベル付き分割を求める．最適なラベル付き分割とは，式（1）で与えられる同時確率が最大になるような $b_{1:K}$ および $y_{1:K}$ である．本研究では，最適なラベル付き分割の探索を，図 2 に示すラティス上の最短経路探索問題に帰着させる．このラティスは，トークン列を左から読んでいくとき，各トークンについてセグメントを継続するか終了するかを選択する過程に基づいている．図に示している「1what」「3end」といった

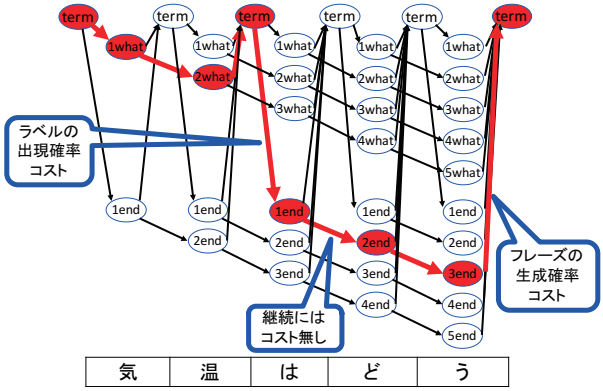

図 2: DPSM の推論を行うための動的計画法のラティス構造

ノードは，対応するトークンが，それぞれ「What スロット」ラベルの 1 単語目,「Ending 非スロット」ラベルの 3 単語目として解釈されることを表す.「term」ノードは，対応するトークン位置でセグメントが終了することを表す.

ラベル付き分割の結果は，このラティス上の経路として表現される．図で太い赤矢印で示された経路は,「気温」が「What スロット」ラベルのフレーズ,「はどう」が「Ending 非スロット」ラベルのフレーズ，と解釈する解析結果を示している．ラティス上の各エッジを通過するのにかかるコストを以下のように対応づけることで，DPSM における尤度最大の分割を行う問題は，ラティス上の最短経路を探索する問題に帰着される．

- セグメントを継続する場合には遷移コストはかからない．
- 「term」ノードへの遷移を行う際には，それまで継続中だったセグメントの確率を言語モデルで計算する．この値の負の対数をコストとする．
- 「term」ノードから別のノードに遷移を行う際には，遷移先のノードが表すラベルの確率を求める．この値の負の対数をコストとする．

これに基づいて，ダイクストラ法を用いて最短経路を求める．

3. 実験

提案手法の有効性を検証するため，実験を行なった．スロットフィリングのデータセットとして，DSTC コーパスと Weather コーパスを用いる．DSTC コーパスは，Dialog State Tracking Challenge 3 [Henderson 15] で提供された英語によるレストラン検索の対話データから，各対話の最初の発話のみを文として収集したものである．スロット情報が一つも含まれていない文は取り除き，文に現れる文字列と表記が異なるスロット値がある場合は手作業で表記を一致させて用いた．DSTC コーパスは 1,441 の文から構成され，**area**, **food**, **price range**, **type**, **children allowed** の 5 種類のスロットを含む．Weather コーパスは，日本語による天気の問い合わせ発話データである．これはホンダ・リサーチ・インスティチュート・ジャパンが収集した非公開のデータであり，プロトタイプのチャット対話システムに対して社内のユーザが行なった発話の履歴のうち，天気情報について問い合わせた文を収集し，人手でスロット情報を付与したものである．Weather コーパスは 1,442 の文から構成され，**when**, **where**, **what** の 3 種類のスロットを含む．トークンの単位は，DSTC コーパスでは単語，Weather コーパスでは文字とした．これは，英語ではスペースで区切られた箇所以外でセグメントが分割されることは考えにくいが，日本語の場合はそのような保証がないためである．

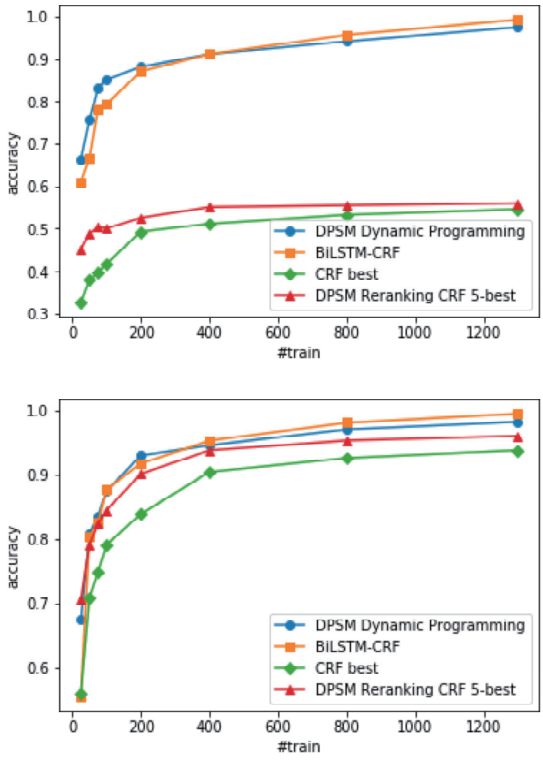

図 3: 上図が Weather コーパス（日本語），下図が DSTC コーパス（英語）のスロット推定精度．横軸は学習データの数，縦軸はテストデータの完全一致正解率を示している．

10 分割交差検定により，推定精度を計測した．また，交差検定におけるそれぞれの学習･テストセットにおいて，学習に用いるデータの数（#train）を減らした時の精度の変化も計測する．比較手法として，条件付確率場（**CRF**），Bidirectional LSTM に CRF を組み合わせたニューラルネットワーク（**BiLSTM-CRF**）[Reimers 17] ，および CRF の 5-best を DPSM によってリランキングする手法（**DPSM Reranking CRF 5-best**）[Wakabayashi 16] を用いる．提案手法は，学習データを用いて DPSM のワンパス学習を行い，動的計画法によってスロットの推定を行う（**DPSM Dynamic Programming**）．

DPSM は Java で実装し，BiLSTM-CRF は Chainer（version 3.3）を用いて Python で実装した．実験は，Ubuntu 16.04，Xeon E5-2660 2.00GHz（14 コア）2 基，メモリ 64GB のサーバ上で実行した．DPSM は並列計算を行わないため，1 スレッドで実行した．BiLSTM-CRF の学習は CPU で行い，物理的に並列して実行可能な 56 スレッドで実行した．

スロットの推定精度は，テストデータに含まれる文のうち，推定されたスロットが完全一致している文の数の割合として求める．図 3 に，スロット推定精度の結果を示す．横軸が学習に用いるデータの数であり，縦軸が 10 分割交差検証によって得られた推定精度の平均値である．結果を見ると，提案手法は深層学習手法とほぼ同等の精度を達成できていることが確認できる．Weather コーパスでは，学習データ数が少ない時には提案手法が最も精度が高く，学習コーパス構築の初期段階から安定した推定を行えることが示唆される．

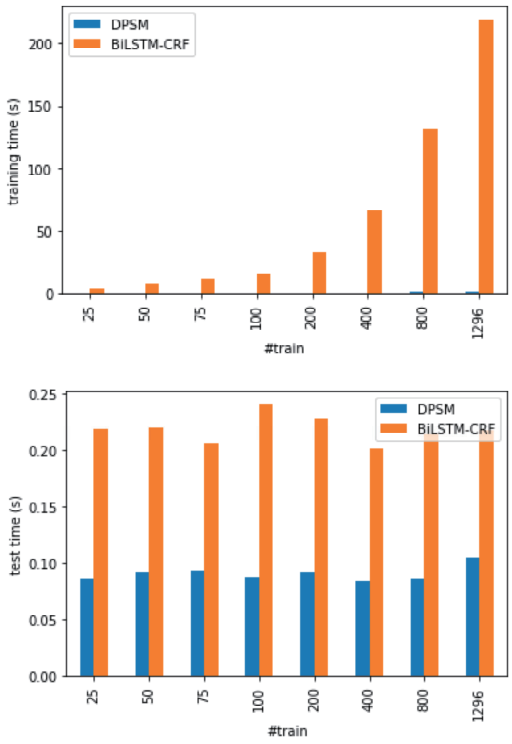

図 4: 学習時（上図）とテスト時（下図）の実行時間の比較

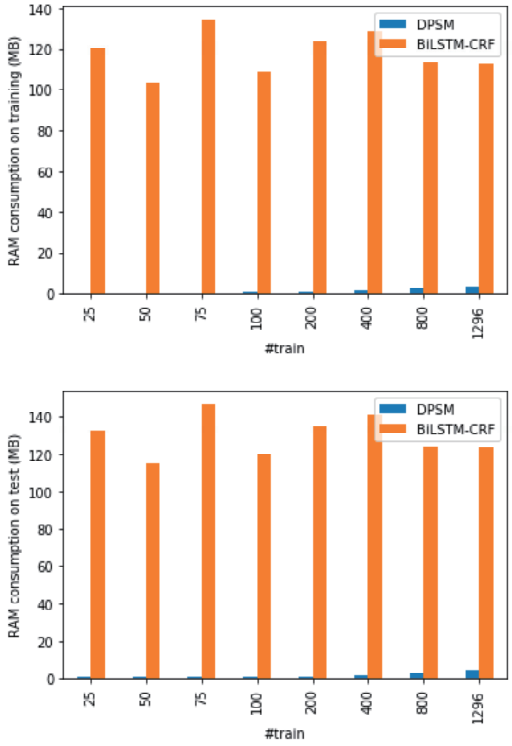

図 5: 学習時（上図）とテスト時（下図）の消費メモリの比較

図 4 に，Weather コーパスにおける学習時とテスト時の実行時間の比較を示す．DSTC コーパスの結果は，Weather コーパスの結果と同様の傾向であるため，紙面の都合上割愛する．横軸は学習データの数，縦軸は 5 回の試行による実行時間の平均値である．学習時の結果を見ると，深層学習手法は学習データが増えるにつれて実行時間が非常に大きくなっているのに対して，提案手法は 1,000 件程度のデータに対しても数秒以内に学習が終了しており，高速であることが分かる．テスト時はどちらの手法も十分高速であるが，提案手法は実行時間が半分程度で済むことがわかる．

図 5 に，Weather コーパスにおける学習時とテスト時の消費メモリ量の比較を示す．実行時間と同様，DSTC コーパスの結果は傾向が同じであるため割愛する．この結果を見ると，学習時，テスト時ともに提案手法の消費メモリ量は深層学習手法と比較して圧倒的に少ないことがわかる．これは，車載端末や携帯端末での利用を考えた時には有用な特性である．

4. 結論

本研究では，言語モデルによる文の最適分割に基づくスロットフィリングのアプローチを提案した．実験により，提案手法は深層学習手法と同等の精度を達成しつつ，高速かつ省メモリであることを示した．学習データを持続的に獲得できる状況では，深層学習では再学習にかかる時間が増加していくため，モデルの更新が追いつかなくなる懸念があるが，提案手法のパラメータ学習はワンパスアルゴリズムであることから，学習データの追加に対しては追加分の計算コストしかかからない．このため，ユーザから得られる教示情報を，即時的に推定結果に反映させるような運用も可能であると考えられる．

参考文献

[Henderson 15] Henderson, M. S.: *Discriminative Methods for Statistical Spoken Dialogue Systems*, PhD thesis, University of Cambridge (2015)

[Mesnil 15] Mesnil, G., et al.: Using Recurrent Neural Networks for Slot Filling in Spoken Language Understanding, *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, Vol. 23, No. 3, pp. 530–539 (2015)

[Reimers 17] Reimers, N. and Gurevych, I.: Reporting Score Distributions Makes a Difference: Performance Study of LSTM-networks for Sequence Tagging, in *Proc. EMNLP* (2017)

[Teh 05] Teh, Y. W., Jordan, M. I., Beal, M. J., and Blei, D. M.: Hierarchical Dirichlet Processes, *Journal of the American Statistical Association*, Vol. 101, pp. 1566–1581 (2005)

[Wakabayashi 16] Wakabayashi, K., Takeuchi, J., Funakoshi, K., and Nakano, M.: Nonparametric Bayesian Models for Spoken Language Understanding, in *Proc. EMNLP* (2016)

[Xu 13] Xu, P. and Sarikaya, R.: Convolutional neural network based triangular CRF for joint intent detection and slot filling, in *Proc. IEEE Workshop on Automatic Speech Recognition and Understanding* (2013)

[Zhai 13] Zhai, K. and Boyd-graber, J.: Online Latent Dirichlet Allocation with Infinite Vocabulary, in *Proc. ICML* (2013)