dishPAM: A Distributable Seeded Hierarchical Pachinko Allocation Model

豊田 樹生　土沢 誉太　築地 毅　菅原 晃平　野口 正樹
ヤフー株式会社
{itoyota, ytsuchiz, ttsukiji, ksugawar, manoguch}@yahoo-corp.jp

1 はじめに

トピックモデリングとタクソノミーの統合に関する研究は近年盛んに取り組まれている [1, 3, 9]．しかし，文書数が巨大な場合やウェブ検索クエリに対するエンティティリンキングのような応用タスクに重点を置いたモデリング及び実用上での評価はまだ十分に行われているとは言い難い．そこで，本研究では hPAM Model 2[8] の派生的なモデルである ***Di**stributable **S**eeded **H**ierarchical **P**achinko **A**llocation **M**odel* (dishPAM) を提案し，次のような貢献を行う:

(i) メトロポリスヘイスティングス法 [7] による最適化ができることを示す．またタクソノミーを用いた教師なし学習によるシード単語 [5] の生成を行い，これを利用したトピック初期化方法を提案する．

(ii) 文書数 (i.e., エンティティ数[1]) が巨大でも対応できるよう，分散処理フレームワーク Apache Spark 上で実装を行った．

(iii) 学習速度，単語予測性能，クラスタリングの安定性の観点から既存手法と比較を行い，高い性能を示したことを報告する．

(iv) 応用タスクとしてウェブ検索クエリに対するエンティティリンキングに取り組み，既存手法と比べ高い適合率を示したことを報告する．

2 提案手法

2.1 dishPAM の単語生成過程

提案手法の dishPAM は hPAM Model 2[8] に対し次の拡張を行った: 1) サブトピックでの単語生成は図 1 に示すように親のスーパートピックごとに異なる分布を用いる．2) ルートトピックでの単語生成は行わない[2]．

単語生成過程は次の通りである:

[1]本研究ではクリックログから各エンティティと対応する文書を生成する

[2]ルートトピックで単語生成を行うか否かで性能に有意差がなかったという報告がある [2]

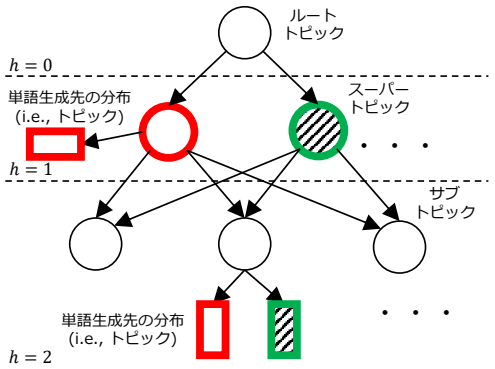

図 1: 単語生成過程の概略図．親のスーパートピックと単語生成先の分布を塗りつぶしパターン及び色で対応付けた．

1. 各々の文書 d において:
   (a) スーパートピックの分布 θ_0 をサンプルする．
   (b) サブトピックの分布 θ_λ をサンプルする．
2. 文書 d の単語 w において:
   (a) 分布 θ_0 からスーパートピック z_λ をサンプルする．
   (b-1) 階層の深さ $h = 1$ で単語生成される場合:
      i. 分布 ϕ_{z_λ} から単語 w をサンプルする．
   (b-2) 階層の深さ $h = 2$ で単語生成される場合:
      i. 分布 θ_λ からサブトピック $z_{\lambda'}$ をサンプルする．
      ii. 分布 $\phi_{z_{\lambda,\lambda'}}$ から単語 w をサンプルする．

上記の単語生成過程に従い，予測分布 p は次のように表現される:

$$p(z_\lambda, z_{\lambda'}, h|\text{rest}) \propto \begin{cases} \hat{\theta_0} \cdot \hat{\phi_{z_\lambda}} & h = 1 \\ \hat{\theta_0} \cdot \hat{\theta_\lambda} \cdot \hat{\phi_{z_{\lambda,\lambda'}}} & h = 2 \end{cases} \quad (1)$$

$$\hat{\theta_0} = \frac{n_{\lambda_j,d} + \alpha_{\lambda_j,d}}{\sum_{j'} \left(n_{\lambda_{j'},d} + \alpha_{\lambda_{j'},d}\right)}, \quad \hat{\theta_\lambda} = \frac{n_{\lambda_j,\lambda'_k,d} + \alpha_{\lambda_j,\lambda'_k,d}}{\sum_{k'} \left(n_{\lambda_j,\lambda'_{k'},d} + \alpha_{\lambda_j,\lambda'_{k'},d}\right)}$$

$$\hat{\phi_{z_\lambda}} = \frac{n_{\lambda_j,w} + \beta_w}{\sum_{w'} \left(n_{\lambda_j,w'} + \beta_{w'}\right)}, \quad \hat{\phi_{z_{\lambda,\lambda'}}} = \frac{n_{\lambda_j,\lambda'_k,w} + \beta_w}{\sum_{w'} \left(n_{\lambda_j,\lambda'_k,w'} + \beta_{w'}\right)}$$

ここで，本論文での主要な記号を表 1 に示す．

表 1: 主要な記号

記号	説明
d	文書
w	単語
h	単語生成時の階層の深さ (0: ルート, 1: スーパートピック, 2: サブトピック)
$n_{\lambda_j,d}$	文書-トピック間でスーパートピック λ_j を経由する回数
$n_{\lambda_j,\lambda'_k,d}$	文書-トピック間でスーパートピック λ_j，サブトピック λ'_k を経由する回数
$n_{\lambda_j,w}$	単語-トピック間でスーパートピック λ_j を経由する回数
$n_{\lambda_j,\lambda'_k,w}$	単語-トピック間でスーパートピック λ_j，サブトピック λ'_k を経由する回数
$\alpha_{\lambda_j,d}$	ルート-スーパートピック λ_j 間のディリクレ分布のパラメータ
$\alpha_{\lambda_j,\lambda'_k,d}$	スーパートピック λ_j-サブトピック λ'_k 間のディリクレ分布のパラメータ
β_w	トピックのディリクレ分布のパラメータ
$\lvert\Lambda\rvert$	スーパートピックの数
$\lvert\Lambda'\rvert$	サブトピックの数
s	単語 w に対して MH 法による遷移前に割り当てられたトピック $s \in \{(a,b) \lvert a \in \{\lambda_0, ..., \lambda_{\lvert\Lambda\rvert-1}\}, b \in \{\text{None}, \lambda'_0, ..., \lambda'_{\lvert\Lambda'\rvert-1}\}\}$
t	単語 w に対する MH 法による遷移先の候補のトピック $t \in \{(a,b) \lvert a \in \{\lambda_0, ..., \lambda_{\lvert\Lambda\rvert-1}\}, b \in \{\text{None}, \lambda'_0, ..., \lambda'_{\lvert\Lambda'\rvert-1}\}\}$

2.2 メトロポリスヘイスティングス法による学習

次の二点について，LightLDA[15] と同様の枠組みで学習を行う: 1) メトロポリスヘイスティングス法を採用する 2) 文書からの提案分布及び単語からの提案分布を設計し，これらの分布を交互に用いて最適化を行う．ただし，dishPAM では上記に加えタクソノミーを用いた教師なし学習によりシード単語 [5] を生成し，これを利用したトピック初期化を行う．

2.2.1 文書からの提案分布

予測分布 p の文書側 (i.e., $\hat{\theta_0}$ 及び $\hat{\theta_\lambda}$ により構成される項) をそのまま用いる場合，疎な項と密な項の二項への分解が困難になる．そこで文書からの提案分布は非正規化を行った近似分布により表現される:

$$p_d(\lambda, \lambda') \propto \begin{cases} n_{\lambda_j,d} + \alpha_{\lambda_j,d} & h = 1 \\ (n_{\lambda_j,d} + \alpha_{\lambda_j,d}) \cdot \left(n_{\lambda_j,\lambda'_k,d} + \alpha_{\lambda_j,\lambda'_k,d}\right) & h = 2 \end{cases} \quad (2)$$

疎な項と密な項への分解　非正規化された分布を用いて，あるトピックが生成される確率を疎な項と密な項の二項に分解した形で表現する．その際，次の手順により，エイリアステーブル [15] を用いてトピックのサンプリングのコストを削減する:

1. 疎な項に対応するエイリアステーブル (分布 ζ_{u_d} と対応)，密な項に対応するエイリアステーブル (分布 ζ_{v_d} と対応) を事前に生成する
2. カテゴリカル分布 $\mathrm{Cat}\left(K=2, \boldsymbol{p}=\left(\frac{u_d}{u_d+v_d}, \frac{v_d}{u_d+v_d}\right)\right)$ を用いて分布 ζ_{u_d} と ζ_{v_d} のどちらを用いるか選択する
3. 選択された分布を用いてトピックをサンプリングする

ここで，u_d, v_d はそれぞれ次のように表現される:

$$u_d = \sum_{j,k}(n_{\lambda_j,d} \cdot n_{\lambda_j,\lambda'_k,d} + n_{\lambda_j,d} \cdot \alpha_{\lambda_j,\lambda'_k,d} + \alpha_{\lambda_j,d} \cdot n_{\lambda_j,\lambda'_k,d}) + \sum_j n_{\lambda_j,d}$$
$$v_d = \sum_{j,k} \alpha_{\lambda_j,d} \cdot \alpha_{\lambda_j,\lambda'_k,d} + \sum_j \alpha_{\lambda_j,d} \quad (3)$$

2.2.2 単語からの提案分布

単語からの提案分布は予測分布 p の単語側 (i.e., $\hat{\phi_{z_\lambda}}$ 及び $\hat{\phi_{z_{\lambda,\lambda'}}}$ により構成される項) を用い，次のように表現される:

$$p_w(\lambda, \lambda') \propto \begin{cases} \frac{n_{\lambda_j,w} + \beta_w}{\sum_{w'}\left(n_{\lambda_j,w'} + \beta_{w'}\right)} & h = 1 \\ \frac{n_{\lambda_j,\lambda'_k,w} + \beta_w}{\sum_{w'}\left(n_{\lambda_j,\lambda'_k,w'} + \beta_{w'}\right)} & h = 2 \end{cases} \quad (4)$$

疎な項と密な項への分解　文書からの提案分布と同様の手順により項の分解及びトピックのサンプリングを行う:

1. 疎な項に対応するエイリアステーブル (分布 ζ_{u_w} と対応)，密な項に対応するエイリアステーブル (分布 ζ_{v_w} と対応) を事前に生成する
2. カテゴリカル分布 $\mathrm{Cat}\left(K=2, \boldsymbol{p}=\left(\frac{u_w}{u_w+v_w}, \frac{v_w}{u_w+v_w}\right)\right)$ を用いて分布 ζ_{u_w} と ζ_{v_w} のどちらを用いるか選択する
3. 選択された分布を用いてトピックをサンプリングする

ここで，u_w,v_w はそれぞれ次のように表現される:

$$u_w = \sum_j \frac{n_{\lambda_j,w}}{\sum_{w'}\left(n_{\lambda_j,w'} + \beta_{w'}\right)} + \sum_{j,k} \frac{n_{\lambda_j,\lambda'_k,w}}{\sum_{w'}\left(n_{\lambda_j,\lambda'_k,w'} + \beta_{w'}\right)}$$
$$v_w = \sum_j \frac{\beta_w}{\sum_{w'}\left(n_{\lambda_j,w'} + \beta_{w'}\right)} + \sum_{j,k} \frac{\beta_w}{\sum_{w'}\left(n_{\lambda_j,\lambda'_k,w'} + \beta_{w'}\right)} \quad (5)$$

2.2.3 受理率

$s \to t$ とトピックが遷移するか否かの受理率 π は次のように表現される:

$$\pi = \begin{cases} \min\left[1.0, \frac{p_w(s)p(t)}{p_w(t)p(s)}\right] & \text{単語からの場合} \\ \min\left[1.0, \frac{p_d(s)p(t)}{p_d(t)p(s)}\right] & \text{文書からの場合} \end{cases} \quad (6)$$

ただし，受理率の計算の際，トピック s に割り当てられている対象の単語は除外して計算を行う．

2.2.4 シード単語を利用したトピック初期化

次の手順によりシード単語を利用したトピック初期化を行う．

はじめに，エンティティの周辺語[3]から構成される仮想文書 [16] を生成する．

つぎに，教師なし学習によりシード単語-クラス-トピック対応表を生成する．クエリログで生起する周辺語を対象として，教師なし学習によるスコアである S_{freq}[10] に基づきクラスごとのランキングを算出し，さらに分布を保持する．この分布は各仮想文書に対してオーバーサンプリングを行い語彙を補う場合に用い

[3] クエリ中のエンティティターム以外の単語

る[4]．また，ランキング上位 5 位以内の周辺語をシード単語と判定し，シード単語-クラス-トピックの対応表を生成する．

さいごに，この対応表を利用した初期トピックの割り当てを行う．手順は次のとおりである: 1) スーパートピックの数 $|\Lambda|$ を最下層のクラスの数と一致させる[5]2) dishPAM の学習の初期化時，各仮想文書の各単語に対して前述の対応表に従いトピックを割り当てる[6]．対応表にエントリがない場合，ランダムにトピックを割り当てる．

3 評価実験

3.1 比較手法

dishPAM(FULL) Spark GraphX 2.2 を利用して実装．パラメータ α は非対称に設定[7]．パラメータ $\beta = 0.01$，$|\Lambda| = 78$，$|\Lambda'| = 117$

dishPAM(-SS) dishPAM(FULL) からタクソノミーを利用した初期トピックの割り当てを省いた手法

Online LDA Online LDA [4] の Spark MLLib 2.2 の実装を利用．トピック数は 500[8]．その他のパラメータはデフォルト．

3.2 データセット

クエリログ 2018 年 9 月の一か月にヤフー検索で発行されたクエリログを使用．

クリックログ 2019 年にヤフー検索で発行されたクリックログを利用．クリック先には Wikipedia, Amazon(書籍のみ), Yahoo!映画, Yahoo!ロコ, メディア系エンティティの公式サイトが含まれる．

タクソノミー及び知識ベース 2019/11/03 付けのタクソノミー及び統合的知識ベース [14] を利用し，クリックログの URL をエンティティに変換した．

[4]各仮想文書に対するサンプリング数の上限は $\min(500, \max(10, N_d))$ と経験的に定めた．ここで，N_d は仮想文書 d の延べ単語数である．

[5]実際のエンティティで最下層のクラスとして用いられていれば，それはスーパートピックのクラスとして算入される．例えば *Thing* → *Place* → *ArchitecturalStructure* に属するエンティティ A と *Thing* → *Place* に属するエンティティ B が存在する場合は，*ArchitecturalStructure* と *Place* の両方ともスーパートピックに算入される．

[6]Jagarlamudi ら [5] とは異なり，初期化時に利用するのみで，hPAM Model 2 の単語生成過程に変更は加えない．

[7]Wallach ら [12] の手法に従い，トピックの規模に応じて値の高低が変化するようにした．

[8]トピック数 78(= $|\Lambda|$)．195(= $|\Lambda| + |\Lambda'|$)．500 の 3 通りで計測した．紙面の都合のため．ヘルドアウトパープレキシティの性能の最も高かったトピック数 500 の結果を報告する．

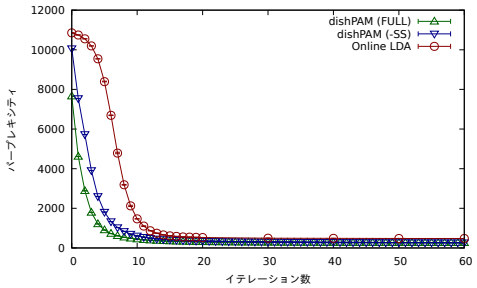

図 2: D_{train} の訓練時のパープレキシティ．各点について 3 回試行した平均，最大値，最小値を記した

仮想文書 表 2 に規模を示す．D_{train} を各手法の訓練に用いた．D_{testE} をヘルドアウトパープレキシティの計測に用いた．また，D_{testT} をエンティティリンキング性能の計測に用いた．ここで，D_{train} と D_{testE} はエンティティの重複がないものとする．

表 2: 仮想文書の規模

仮想文書	クリックログの期間	文書数	異なり単語数	延べ単語数	オーバーサンプル
D_{train}	2019/01/01~10/31	462,874	533,127	103,948,457	有
D_{testE}	2019/01/01~10/31	116,025	330,525	25,964,003	有
D_{testT}	2019/11/01~11/30	149,440	67,857	6,126,898	無

3.3 基本的指標の比較

学習速度，単語予測性能，クラスタリングの安定性の観点から既存手法と比較を行った．

学習速度の比較 学習速度の比較のため，D_{train} を用いて訓練を行った際のイテレーション毎のパープレキシティの変化を図 2 に示す．dishPAM(FULL) 及び dishPAM(-SS) は Online LDA に対して収束までのイテレーション回数が少ないことがわかる．また，dishPAM(FULL) と dishPAM(-SS) を比較すると dishPAM(FULL) のほうが少ないイテレーション回数で収束しており，対応表を利用した初期トピック割り当ての収束速度に対する効果が観測できた．

単語予測性能の比較 単語予測性能の比較を行うためヘルドアウトパープレキシティ [13] の計測を行った．D_{train} を用いて訓練を行い D_{testE} に対して $\exp(-\frac{\sum_{d \in D_{\text{testE}}} \ln P(d|\mathcal{M})}{\sum_{d \in D_{\text{testE}}} N_d})$ の計測を行った[9]．表 3 にその結果を示す．ここで，$\mathcal{M}$ はモデル，N_d は文書 d の延べ単語数である．

dishPAM(FULL) 及び dishPAM(-SS) は Online LDA の値の約 41%程度までパープレキシティが低下

[9]document completion[13] に分類される手法を用いた．60 回イテレーションを行い θ_{test} を推定したのち，$\phi_{\text{train}}, \theta_{\text{test}}$ を用いてパープレキシティを計測した．

し大きな改善がみられた．また，dishPAM(FULL) は dishPAM(-SS) に対してパープレキシティが約 5 低下した．このことから，対応表を利用した初期トピック割り当てがわずかながらも単語予測性能の改善に寄与したといえる．

クラスタリングの安定性の比較　クラスタリングの安定性を比較するため Variation of Information(VI)[6, 11] の計測を行った．D_{train} の訓練を各手法ごとにそれぞれ 3 回行い，形成されたモデル間での VI を計測した．結果を表 3 に示す．

dishPAM(FULL) 及び dishPAM(-SS) は Online LDA の値の約 6%程度まで VI が低下し非常に高い安定性をみせた．また，dishPAM(FULL) は dishPAM(-SS) に対して VI が約 0.002 低下した．このことから，対応表を利用した初期トピック割り当てがわずかながらも安定性の改善に寄与したといえる．

3.4　エンティティリンキング性能の比較

応用的なタスクとしてウェブ検索クエリに対するエンティティリンキング性能の比較を行った．

評価用事例の作成　次の手順で評価用事例を作成した: 1) D_{testT} のそれぞれのエンティティに対して，名前として正式名称もしくは別名を付与する．2) 名前の一致するエンティティ同士をグルーピングする 3) 各グループにおいて，ある周辺語がグループ内の一つのエンティティのみに生起している場合，その周辺語とエンティティのペアを正例とする．また，グループ内の残りのエンティティについて，その周辺語とのペアを負例とする．4) グループの正例と負例の合計数が 3 未満であるようなグループを除外する．

結果，評価用事例となるグループが 70,211 件生成された．

エンティティリンキング性能の比較　各グループについて，エンティティと周辺語のペアをモデルに対して与えた際に，正例の生成確率を候補間で最大にできているか否かを測定した．図 3 に結果を示す．適合率が約 0.865 を下回るような区間では Online LDA が dishPAM(FULL) の再現率を上回った．一方，適合率が約 0.865 を超えるような区間のほとんどの同適合率点において，dishPAM(FULL) は Online LDA に対して再現率で上回っていた．また，11 点平均適合率は dishPAM(FULL) が約 0.925, Online LDA が約 0.917 となり，dishPAM(FULL) の高い適合率が示された．

表 3: ヘルドアウトパープレキシティ及び Variation of Information(VI) の比較．値は 3 回試行した平均．値が低いほうが性能が良い．

モデル	ヘルドアウトパープレキシティ	VI
dishPAM(FULL)	**305.725** ±0.146	**0.021** ±0.000
dishPAM(-SS)	310.615 ±1.009	0.023 ±0.000
Online LDA	757.905 ±0.811	0.380 ±0.001

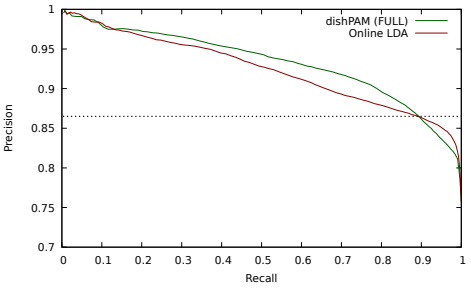

図 3: 再現率-適合率@1 グラフ．点線は適合率 0.865 点．生成確率 (対数) に対して $[-18.0, -0.5]$ の区間で閾値を設定．

4　おわりに

本研究ではトピックモデリングとタクソノミーを組み合わせたフレームワークである dishPAM を提案した．基本的な評価指標及びエンティティリンキングタスクでの性能を既存手法と比較し，高い性能を達成したことを報告した．

参考文献

[1] Anton Bakalov, Andrew McCallum, Hanna Wallach, and David Mimno. Topic models for taxonomies. In *Proceedings of the 12th ACM/IEEE-CS joint conference on Digital Libraries*, pp. 237–240. ACM, 2012.

[2] Chaitanya Chemudugunta, Padhraic Smyth, and Mark Steyvers. Modeling general and specific aspects of documents with a probabilistic topic model. In *Advances in neural information processing systems*, pp. 241–248, 2007.

[3] Chaitanya Chemudugunta, Padhraic Smyth, and Mark Steyvers. Combining concept hierarchies and statistical topic models. In *Proceedings of the 17th ACM conference on Information and knowledge management*, pp. 1469–1470. ACM, 2008.

[4] Matthew Hoffman, Francis R Bach, and David M Blei. Online learning for latent dirichlet allocation. In *advances in neural information processing systems*, pp. 856–864, 2010.

[5] Jagadeesh Jagarlamudi, Hal Daumé III, and Raghavendra Udupa. Incorporating lexical priors into topic models. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 204–213. Association for Computational Linguistics, 2012.

[6] Marina Meilă. Comparing clusterings by the variation of information. In *Learning theory and kernel machines*, pp. 173–187. Springer, 2003.

[7] Nicholas Metropolis, Arianna W Rosenbluth, Marshall N Rosenbluth, Augusta H Teller, and Edward Teller. Equation of state calculations by fast computing machines. *The journal of chemical physics*, Vol. 21, No. 6, pp. 1087–1092, 1953.

[8] David Mimno, Wei Li, and Andrew McCallum. Mixtures of hierarchical topics with pachinko allocation. In *Proceedings of the 24th international conference on Machine learning*, pp. 633–640. ACM, 2007.

[9] Viet-An Nguyen, Jordan L Ying, Philip Resnik, and Jonathan Chang. Learning a concept hierarchy from multi-labeled documents. In *Advances in Neural Information Processing Systems*, pp. 3671–3679, 2014.

[10] Marius Paşca, Benjamin Van Durme, and Nikesh Garera. The role of documents vs. queries in extracting class attributes from text. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, pp. 485–494. ACM, 2007.

[11] Hanna M Wallach and David Mimno Andrew McCallum. Supplementary materials for "rethinking lda: Why priors matter".

[12] Hanna M Wallach, David M Mimno, and Andrew McCallum. Rethinking lda: Why priors matter. In *Advances in neural information processing systems*, pp. 1973–1981, 2009.

[13] Hanna M Wallach, Iain Murray, Ruslan Salakhutdinov, and David Mimno. Evaluation methods for topic models. In *Proceedings of the 26th annual international conference on machine learning*, pp. 1105–1112. ACM, 2009.

[14] Tomoya Yamazaki, Kentaro Nishi, Takuya Makabe, Mei Sasaki, Chihiro Nishimoto, Hiroki Iwasawa, Masaki Noguchi, and Yukihiro Tagami. A scalable and plug-in based system to construct a production-level knowledge base. In *Proceedings of the 1st International Workshop on Challenges and Experiences from Data Integration to Knowledge Graphs co-located with the 25th ACM SIGKDD International Conference on Knowledge Discovery &Data Mining*, 2019.

[15] Jinhui Yuan, Fei Gao, Qirong Ho, Wei Dai, Jinliang Wei, Xun Zheng, Eric Po Xing, Tie-Yan Liu, and Wei-Ying Ma. Lightlda: Big topic models on modest computer clusters. In *Proceedings of the 24th International Conference on World Wide Web*, pp. 1351–1361. International World Wide Web Conferences Steering Committee, 2015.

[16] 豊田樹生, 夜久真也, 石川葉子, 土沢誉太, Kulkarni Kaustubh, Bhattacharjee Anupam, 宰川潤二. ウェブ検索クエリに対する周辺語を考慮した教師なしエンティティリンキング. 言語処理学会第 25 回年次大会発表論文集, pp. 81–84, 2019.