

オンライン広告におけるウェブ閲覧系列の分散表現の獲得

Obtaining Distributed Representations of Web Browsing Sequences for Online Advertising

田頭 幸浩^{*1} 小林 隼人^{*1} 小野 真吾^{*1} 田島 玲^{*1}
Yukihiro Tagami Hayato Kobayashi Shingo Ono Akira Tajima

^{*1}ヤフー株式会社
Yahoo Japan Corporation

Large scale user modeling, based on the user activities on the Web, plays a key role in online advertising targeting. In our previous study, we introduced an approach that summarizes each sequence of user Web page visits using the Paragraph Vector, considering users and URLs as paragraphs and words, respectively. The learned user representations are used among the user-related prediction tasks in common. In addition, on the basis of analysis of our Web page visits data, we proposed Backward PV-DM, which is a modified version of Paragraph Vector. In this paper, we also report experimental results for evaluating the effect of data size.

1. 導入

オンライン広告を限られた時間や予算のもとで効率的に配信するためには、その広告に興味を持ったり、広告主の顧客になりそうなユーザーに対して、優先的に広告を表示することが好ましい。そのため、ウェブ上の行動をもとにユーザーをモデリングし、広告を表示するユーザーを適切に絞り込むことは、非常に重要なタスクである。

近年、自然言語処理の分野では、ベクトル空間上での単語の分散表現が注目を浴びている [Levy 14, Mikolov 13]。従来、それぞれの単語を個々のシンボルとして扱うアプローチが主であったが、これらのアプローチでは単語を固定長の密なベクトルで表現する。これらの、さまざまな学習手法で得られた単語ベクトルは、統語的や意味的な関係を捉えた表現になっており、類推タスクなどでの有効性が示されている [Levy 14, Mikolov 13]。また、パラグラフや文書などの、可変長のテキスト列の分散表現を得る研究もなされている [Le 14]。このアプローチは従来の単語 n-gram や単語ベクトルの単純な平均と比較して、評判分析のタスクにおいて良い結果を示している [Le 14]。

これらの研究を受けて、我々は、Paragraph Vector [Le 14] を用いてウェブ上のユーザーの行動列を集約するアプローチを提案した [Tagami 15]。このアプローチでは、ユーザーをパラグラフもしくは文書、ウェブページ訪問を単語と見なし、ユーザーのウェブページ訪問系列に対してこの自然言語処理の手法を適用する。学習で得られたユーザーを表現する低次元の特徴量ベクトルは、ユーザーに関連した各種予測タスクで共通して用いることができる。さらに、英語版 Wikipedia と Yahoo! JAPAN のアクセスログをもとに作成したウェブ閲覧系列の、二つのデータの分布の違いに着目し、Paragraph Vector を修正した Backward PV-DM を提案し、その有用性について確かめた [Tagami 16]。

上記のアプローチでは、ユーザーのベクトル表現をログから教師無し学習で得ることができるため、ウェブサイトやスマートフォンアプリ全体ではユーザーの行動ログが潤沢に得られる一方で、個々の予測タスクの学習データが少ない場合に、このベクトル表現は有効な特徴量になると期待される。本稿では、まず我々のアプローチの概要を述べ、続いて学習データ量を変化させた時の、予測タスクの性能の変化について報告する。

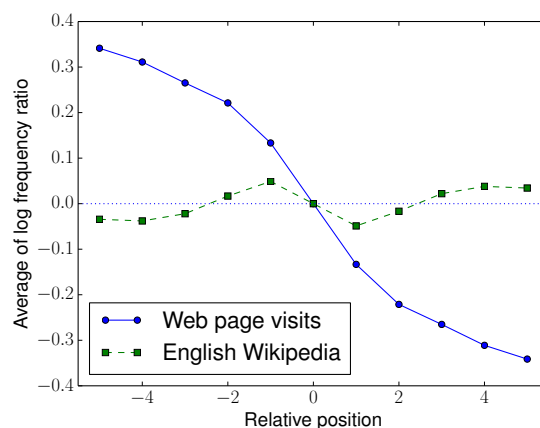


図 1: セッションおよび文中の相対位置に対する、URL および単語の出現頻度の、平均対数比率のプロット。

2. ユーザーのウェブページ訪問系列

まず、可能なユーザー行動の集合を A と定義する。 i 番目のユーザー u_i の、ウェブ上の行動系列を、以下のように定義する。

$$(a_{i,1}, a_{i,2}, \dots, a_{i,T_i})$$

ただし、 $a_{i,t} \in A$ はユーザー u_i の t 番目の行動を、 T_i はその系列の長さを表す。

本稿では、ユーザーの行動のうちウェブページ訪問に注目し、各ページ訪問 $a_{i,t}$ をそのウェブページの URL で表現する。この URL はウェブサービスのアクセスログから抽出することで容易に獲得することができる。以下では、“ウェブページ訪問”と“URL”を区別せずに用いる。

自然言語処理ベースのアプローチを適用するため、Yahoo! JAPAN のアクセスログからウェブ閲覧系列のデータを獲得し、これを英語版 Wikipedia のデータと比較した [Tagami 16]。これらの二つのデータを比較することにより、以下のような知見が得られた。

- 類似性: ウェブページ閲覧データに含まれる URL の頻度は、べき分布に従う。広く知られているように [Clauaset 09],

英語版 Wikipedia の単語頻度も同じ性質を示した。

- 相違点：一つのセッションや文の中での相対位置に着目すると、二つのデータの頻度分布は著しく異なった。

二つ目の知見に関連して、セッションおよび文中の相対位置に対する、URL および単語の出現頻度の、平均対数比率を図 1 に示す。図から、ウェブページ訪問系列データでは、セッションの後半部に、相対的に出現頻度が少ない URL が出現することが分かる。これらの、セッションの後半部に現れる、相対的に出現頻度が少ない URL は、ユーザーの興味や選好を捉えるために重要な役割を果たすものと考えられる。

3. ベクトルモデル

この章では、ウェブページの訪問系列から、ユーザーのベクトル表現を獲得するモデルと学習法について述べる。

3.1 PV-DM

この節では、対象とする問題設定にしたがって Paragraph Vector [Le 14] のモデルの一つである、PV-DM (Distributed Memory Model of Paragraph Vector) について説明する。先述した通り、ユーザーをパラグラフもしくは文書、ユーザー行動を単語と見なし、このベクトルモデルをユーザーの行動列に対して適用する。 i 番目のユーザー u_i の系列に対して、このベクトルモデルは、以下の対数確率の和を目的関数とし、最大化を行う。

$$\sum_t \log p(a_{i,t} | a_{i,t-1}, \dots, a_{i,t-s}, u_i)$$

なお、 s はコンテキストウィンドウのサイズである。これは、直前の行動 $a_{i,t-1}, \dots, a_{i,t-s}$ とユーザー u_i が与えられた時の、対象の行動 $a_{i,t}$ の条件付き確率を表している。PV-DM はこのマルチクラス問題の確率を、以下のように softmax 関数を用いて、log-bilinear モデルとして定義する。

$$p(a_{i,t} | a_{i,t-1}, \dots, a_{i,t-s}, u_i) := \frac{\exp(\mathbf{w}_{a_{i,t}}^T \mathbf{v}_I)}{\sum_{a \in A} \exp(\mathbf{w}_a^T \mathbf{v}_I)} \quad (1)$$

ここで、 $\mathbf{w}_{a_{i,t}}$ は行動 $a_{i,t}$ に対応する“出力”ベクトルを、 \mathbf{v}_I は対象の時刻以前の行動 $a_{i,t-1}, \dots, a_{i,t-s}$ とユーザー u_i に対応する“入力”ベクトルである。また、行動 $a_{i,t}$ に対応する“入力”ベクトルを $\mathbf{v}_{a_{i,t}}$ 、ユーザー u_i に対応する“入力”ベクトルを \mathbf{v}_{u_i} と定義する。そして、 \mathbf{v}_I は以下のように連結したベクトルとして表現される。

$$\mathbf{v}_I = [\mathbf{v}_{a_{i,t-1}}^T, \dots, \mathbf{v}_{a_{i,t-s}}^T, \mathbf{v}_{u_i}^T]^T$$

$j \leq 0$ となる $\mathbf{v}_{a_{i,j}}$ は特別なベクトル \mathbf{v}_{NULL} で置き換える。

学習で得られたユーザーに対応する“入力”ベクトル \mathbf{v}_{u_i} を、さまざまな予測タスクの特徴量として用いる。

3.2 Backward PV-DM

この節では、英語版 Wikipedia データとウェブページ閲覧系列データの比較で得られた知見をもとに提案された、Backward PV-DM [Tagami 16] について述べる。通常の PV-DM との違いはコンテキストウィンドウである。Backward PV-DM は以下の対数確率の和を目的関数とし、最大化を行う。

$$\sum_t \log p(a_{i,t} | a_{i,t+1}, \dots, a_{i,t+s}, u_i)$$

表 1: *AdClicker* と *SiteVisitor* データセットの統計値。#Features は各データセットに含まれるユニークな URL 数。

Data set	#Train	#Validation	#Test	#Features
<i>AdClicker</i>	51,576	10,000	10,000	66,957
<i>SiteVisitor</i>	1,862,693	20,000	20,000	1,219,850

PV-DM が対象の行動 $a_{i,t}$ を予測するために、以前の行動 $a_{i,t-1}, \dots, a_{i,t-s}$ を用いるのに対し、このモデルは後続の行動 $a_{i,t+1}, \dots, a_{i,t+s}$ を用いる。条件付き確率は以下のように定義される。

$$p(a_{i,t} | a_{i,t+1}, \dots, a_{i,t+s}, u_i) := \frac{\exp(\mathbf{w}_{a_{i,t}}^T \mathbf{v}_I)}{\sum_{a \in A} \exp(\mathbf{w}_a^T \mathbf{v}_I)} \quad (2)$$

$$\mathbf{v}_I = [\mathbf{v}_{a_{i,t+1}}^T, \dots, \mathbf{v}_{a_{i,t+s}}^T, \mathbf{v}_{u_i}^T]^T$$

3.3 ベクトルモデルの学習

一般的に行動の種類数 $|A|$ は多く、式 1 や 2、それらの一階導関数の計算コストは高いため、そのまま計算することは実用的ではない。そのため、Le と Mikolov [Le 14] は単語の頻度を考慮した階層 softmax を用いることで学習の高速化を行った。ここでは階層 softmax の代わりに、negative sampling [Mikolov 13] のアプローチをとる。このアプローチでは、式 2 を代入した目的関数 $\log p(a_{i,t} | a_{i,t+1}, \dots, a_{i,t+s}, u_i)$ を以下の式で近似する。

$$\log \sigma(\mathbf{w}_{a_{i,t}}^T \mathbf{v}_I) + k \cdot \mathbb{E}_{a_n \sim p_n(a)} [\log \sigma(-\mathbf{w}_{a_n}^T \mathbf{v}_I)]$$

なお、 $\sigma(z) = 1/(1 + \exp(-z))$ は sigmoid 関数であり、 k は negative sample の数、 $p_n(a)$ は negative sample を生成する分布である。Mikolov ら [Mikolov 13] に従い、 $p_n(a)$ として単語の頻度分布 $U(a)$ を $3/4$ 乗した分布を用いる。このモデルを AdaGrad [Duchi 11] を用いた非同期 SGD (Asynchronous Stochastic Gradient Descent) [Recht 11] で学習する。新しいユーザーに対する推論時には、行動に対応する入力と出力のベクトル \mathbf{v}_a と \mathbf{w}_a を固定して、ユーザーに対応するベクトル \mathbf{v}_u のみを同様に学習する。

4. 実験

この章では、提案手法の評価実験を行う。まず、データセットと実験設定、比較手法について述べ、その後、実験結果を示し考察を行う。

4.1 データセット

提案手法を評価するため、教師あり学習のデータセットとして *AdClicker* データセットと *SiteVisitor* データセットの二つを用意した。*AdClicker* データセットは、選択された、ある五つの広告キャンペーンに含まれるコンテキスト広告をクリックしたユーザーの集合からなる。同様に、*SiteVisitor* データセットは、選択された、ある五つの広告主のウェブサイトを訪れたユーザーの集合から構成される。

これら二つのデータセットは、前日のウェブページ訪問をもとに、その翌日のユーザーの行動を予測する問題として作成した。学習データとバリデーションデータは 2014 年 7 月 22 日と 23 日のログから作成した。22 日のウェブページ訪問を特徴量として用い、23 日の広告クリックや広告主のサイト訪問をラベルとした。同様に、テストデータは 2014 年 7 月 23 日と

表 2: 実験結果. 値は評価指標である AUC を表す. ベクトルモデルは 5 回の平均値を報告している.

	AdClicker					SiteVisitor				
	Ac1	Ac2	Ac3	Ac4	Ac5	Sv1	Sv2	Sv3	Sv4	Sv5
<i>Bin</i>	0.9753	0.8063	0.6641	0.7052	0.7524	0.7619	0.8188	0.7087	0.7920	0.7292
<i>Freq</i>	0.9814	0.8184	0.6580	0.6961	0.7509	0.7821	0.8163	0.7006	0.7781	0.7256
<i>CBoW</i>	0.9903	0.8323	0.6533	0.7154	0.7700	0.7999	0.8277	0.7067	0.7849	0.7339
<i>Skip-gram</i>	0.9906	0.8354	0.6562	0.7163	0.7725	0.8017	0.8328	0.7135	0.7931	0.7417
<i>Directed Skip-gram</i>	0.9904	0.8374	0.6533	0.7159	0.7706	0.8019	0.8308	0.7120	0.7914	0.7394
<i>PV-DM</i>	0.9899	0.8151	0.6483	0.7242	0.7633	0.8051	0.8343	0.7180	0.7964	0.7479
<i>Backward PV-DM</i>	0.9902	0.8247	0.6537	0.7345	0.7661	0.8092	0.8366	0.7222	0.8028	0.7491
<i>PV-DBoW</i>	0.9894	0.8288	0.6507	0.7290	0.7581	0.7965	0.8294	0.7198	0.7945	0.7489

24 日のログから作成した. 上記の特徴量は Yahoo! JAPAN のウェブサービスのログから抽出したものであるため, ウェブ上でのユーザーの行動のほんの一部であることを注意が必要である. また, *SiteVisitor* データセットのラベルである, 広告主のサイト訪問は含まれない.

AdClicker データセットに含まれるコンテキスト広告は, ユーザーの情報だけでなくウェブページのコンテンツも考慮して表示される. そのため, このタスクにおいては, ウェブページの表現を学習することも有効であると考えられる. 一方で, *SiteVisitor* はより複雑なユーザーの興味に基づいたデータセットである.

一人のユーザーが複数の広告をクリックしたり, さまざまな広告主のサイトを訪問することがあり得るため, これら二つのデータセットはマルチラベルのデータセットである. 実験では, マルチラベル問題を二値分類問題の集合に変換した. *AdClicker* データセットと *SiteVisitor* データセットから得られた二値分類問題を, それぞれ Ac1 から Ac5, Sv1 から Sv5 と表す. それぞれの手法で抽出された特徴量を用いてロジスティック回帰のモデルを学習し, AUC (Area Under ROC Curve) を指標として評価した. 二つのデータセットの統計量を表 1 にまとめた.

4.2 実験設定と比較手法

実験では Paragraph Vector を用いたアプローチといくつかのベースラインを比較した. *Bin* と *Freq* は URL をそのまま特徴量として用いた手法である. *Freq* はユーザーのページ訪問の頻度を考慮するのに対し, *Bin* はウェブページに訪れたか否かのみを用いた. 言い換えると, *Bin* はバイナリ素性, *Freq* は Term Frequency を素性とした場合に対応し, 特徴量は高次元でスパースなベクトルで表現される.

ここでは, CBoW, Skip-gram [Mikolov 13], Directed Skip-gram [Djuric 14] を単語ベクトルモデルと呼ぶ. 同様に, PV-DM, Backward PV-DM, PV-DBoW を Paragraph Vector と呼ぶ. 単語ベクトルモデルを用いる場合, Djuric ら [Djuric 14] が行ったように, ウェブ閲覧系列中に含まれる行動ベクトル \mathbf{v}_a を単純に平均したベクトルで, ユーザーを表現する. Paragraph Vector の場合は, ユーザーベクトル \mathbf{v}_u をユーザーの表現として用いる. ベクトルモデルを用いた手法は, イタリックで表す. 例えば, PV-DM を用いた手法は, *PV-DM* と表す.

ベクトルモデルの学習のために, 2014 年 7 月 22 日のアクセスログの一部を用いて, 各ユーザーが訪れたウェブページの URL を抽出した. このアクセスログには, 通常のウェブサービスの他に, スマートフォンやタブレット端末のアプリケーションによるものも含まれている. ウェブページ訪問数が 10 から 1,000 のユーザーを抽出し, 出現回数が 5 回を下回る URL を削除した. 二つの連続するページ訪問の時刻の差が 30 分以

上の場合, セッションの区切りと見なした. ウェブ閲覧系列中の一セッションは, パラグラフや文章中の一文に対応する. 結果として, 得られたデータ中の URL のユニーク数はおよそ 387 万, ページ訪問の総数は約 10 億となった

ベクトルモデルの学習の設定は, 入力ベクトルのサイズを $v_a = v_u = 400$, コンテキストウィンドウのサイズを $s = 5$, ネガティブサンプルの数を $k = 5$, エポック数 (SGD におけるデータの周回数) を 5 とした. *AdClicker* データセットと *SiteVisitor* データセットの学習データとバリデーションデータは, 教師無し学習のデータと同じく 2014 年 7 月 22 日のログから作成したため, それらのデータに含まれるユーザーのベクトルは Paragraph Vector の学習時に得られるが, ここではテストデータに含まれる未知のユーザーと同様に推論のステップを経てユーザーベクトルを獲得し, 特徴量ベクトルとした.

4.3 実験結果

実験結果を表 2 にまとめた. 太字の項目は, 各手法のうち最大の結果を示している.

SiteVisitor データセットでは *PV-DM* が *Skip-gram* よりも良い結果を示したが, *AdClicker* では逆の傾向が見られた. これは, 先に述べたデータセットの違いに起因したものだと考えられる. 特徴量として URL をそのまま用いる *Bin* と *Freq* は, ほとんどの場合において劣った結果となった.

我々の提案手法である *Backward PV-DM* は, *PV-DM* よりも常に良い結果を示し, *SiteVisitor* データセットにおいては比較した手法の中で最良の結果となった.

4.4 データサイズの影響

はじめに述べたように, 潤沢に得られるユーザーの行動系列を用いることで, 学習データが小規模な予測タスクにおいても良い予測精度を達成したい. 図 2 に, *AdClicker* と *SiteVisitor* の学習データの数を変化させた時の実験結果の一部を示す. URL をそのまま特徴量として用いる *Bin* と *Freq* と比較して, ベクトルモデルは学習データのサイズが少ない場合において良い結果を示した. ベクトルモデルは, 低次元かつ密なベクトルを学習するため, このような場合においては有利であると言える.

さらに, ユーザーを表現するベクトルを学習する際の, ウェブページ訪問系列の数を増加させることによって, 予測タスクの精度が向上するかを確認した. 図 3 に, ベクトルモデルの学習に用いるデータ数を変化させた時の実験結果の一部を示す. *SiteVisitor* では, ウェブページ訪問系列を増加させることによって予測タスクの性能が向上することが, 五つのタスク中四つで確認できた. 一方で, *AdClicker* では, 特に関連性は見られなかった. *SiteVisitor* でみられた向上度合いは, それぞれのタスクではそれほど大きなものではないが, 獲得されたユーザーのベクトル表現は, 各種予測タスクで共通して用いることができるため, 総体としては決して小さなものではないと考えられる.

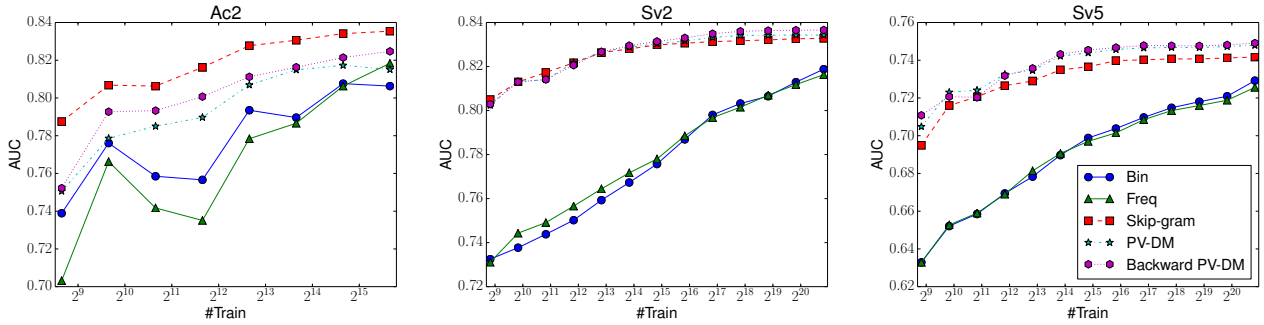


図 2: 予測タスクの学習データの量を変化させた時の実験結果。横軸は対数スケールとなっている。スペースの制限のため一部の結果のみ示している。

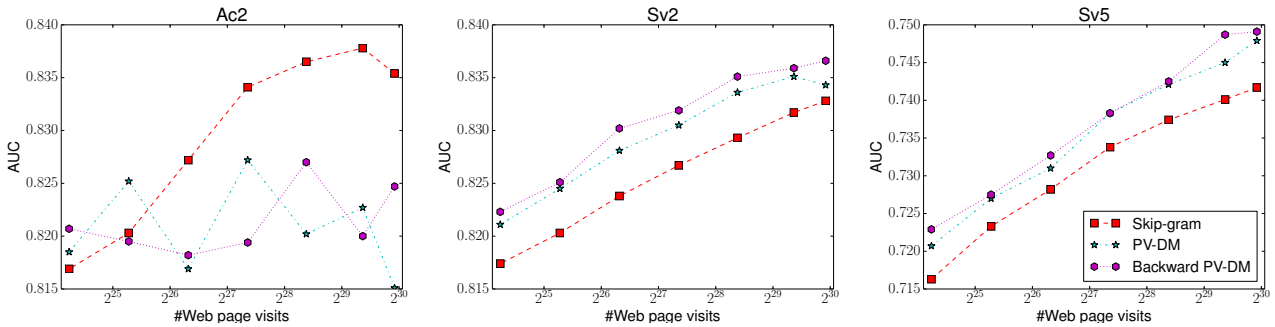


図 3: ベクトル表現を獲得するためのウェブ閲覧の数を変化させた時の実験結果。横軸は対数スケールとなっている。スペースの制限のため一部の結果のみ示している。

5. まとめと今後の課題

本稿では、ユーザーのウェブ閲覧系列から教師無し学習によってユーザー表現を得ることを目的とし、Paragraph Vectorを用いてユーザーの行動列を集約するアプローチについて述べた。Yahoo! JAPAN のログをもとに作成した 2 種類の予測タスクのデータセットを用いて実験を行い、このアプローチを評価した。また、学習データ量を変化させた時の、予測タスクの性能の変化についても示した。

今後の課題としては、三つの方向性が考えられる。本稿では、ユーザーの行動として閲覧したウェブページの URL に着目し、これを元にユーザーの表現を学習したが、検索クエリやウェブページのコンテンツなどのさまざまな情報を用いることで、表現の質が向上することが期待される。また、ユーザーの表現を得る際に、教師なし学習以外の、半教師あり学習やマルチタスク学習、マルチタスク学習などの学習方法も有効ではないかと期待される。最後に、ウェブスケールのユーザーデータに対して、より表現力が高い LSTM [Hochreiter 97] などを用いたモデルを効率的に学習する手法の開発も、今後の課題である。

参考文献

[Clauset 09] Clauset, A., Shalizi, C. R., and Newman, M. E. J.: Power-Law Distributions in Empirical Data, *SIAM Review* (2009)

[Djuric 14] Djuric, N., Radosavljevic, V., Grbovic, M., and Bhamidipati, N.: Hidden Conditional Random Fields with Distributed User Embeddings for Ad Targeting, in *ICDM* (2014)

[Duchi 11] Duchi, J., Hazan, E., and Singer, Y.: Adaptive Subgradient Methods for Online Learning and Stochastic Optimization, *J. Mach. Learn. Res.* (2011)

[Hochreiter 97] Hochreiter, S. and Schmidhuber, J.: Long short-term memory, *Neural computation* (1997)

[Le 14] Le, Q. and Mikolov, T.: Distributed Representations of Sentences and Documents, in *ICML* (2014)

[Levy 14] Levy, O. and Goldberg, Y.: Neural Word Embedding as Implicit Matrix Factorization, in *Advances in Neural Information Processing Systems 27* (2014)

[Mikolov 13] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J.: Distributed Representations of Words and Phrases and their Compositionality, in *NIPS* (2013)

[Recht 11] Recht, B., Re, C., Wright, S., and Niu, F.: Hogwild: A Lock-Free Approach to Parallelizing Stochastic Gradient Descent, in *NIPS* (2011)

[Tagami 15] Tagami, Y., Kobayashi, H., Ono, S., and Tajima, A.: Modeling User Activities on the Web Using Paragraph Vector, in *WWW Companion* (2015)

[Tagami 16] Tagami, Y., Kobayashi, H., Ono, S., and Tajima, A.: Distributed Representations of Web Browsing Sequences for Ad Targeting, in *TargetAd* (2016)