Distributed Representations of Web Browsing Sequences for Ad Targeting

Yukihiro Tagami, Hayato Kobayashi, Shingo Ono, Akira Tajima
Yahoo Japan Corporation
Tokyo, Japan
{yutagami, hakobaya, shiono, atajima}@yahoo-corp.jp

ABSTRACT

Large scale user modeling, based on the user activities on the Web, plays a key role in online advertising targeting. In our work-in-progress paper [15], we introduced an approach that summarizes each sequence of user Web page visits using the Paragraph Vector [8], considering users and URLs as paragraphs and words, respectively. The learned user representations are used among the user-related prediction tasks in common. In this paper, on the basis of analysis of our Web page visits data, we propose Backward PV-DM, which is a modified version of Paragraph Vector. We show experimental results on two ad-related data sets based on logs from Web services of Yahoo! JAPAN. Our proposed method achieved better results than existing vector models.

Categories and Subject Descriptors

H.2.8 [**Database management**]: Database applications—*Data mining*; I.2.6 [**Artificial Intelligence**]: Learning; J.4 [**Social and behavioral sciences**]: Economics

Keywords

Online advertising, Web browsing behavior, Paragraph Vector, representation learning.

1. INTRODUCTION

For efficient advertising, ads should be shown to the users who are interested in them or likely to be customers for each advertiser. Thus sophisticated user modeling for targeting, based on the user activities on the Web, is very important.

Recently, in the natural language processing (NLP) field, distributed representations of words in a vector space have received much attention [10]. The studies that employ this approach represent words as fixed length dense vectors, whereas the conventional approach treats individual words as unique symbols. These vector representations, which are learned with various training methods, capture syntactic and semantic word relationships. In addition, some researchers have proposed models to learn vector representations for variable-length pieces of text such as sentences, paragraphs, and documents [8]. In a sentiment analysis task, this approach achieves better results than the conventional word n-gram model and simple averaging of word vectors.

Following these successful techniques, in our work-in-progress paper [15], we proposed an approach that summarizes each sequence of user Web page visits using the Paragraph Vector [8], which is an unsupervised method that learns continuous distributed vector representations from pieces of text. In other words, we apply the vector model to sequences of user Web page visits, considering users and visits as paragraphs (or documents) and words, respectively. The learned low-dimensional feature vectors are used among the user-related prediction tasks in common.

However, do we simply treat the Web page visits data the same as natural language data? These two types of data are probably generated from different distributions. Therefore, in this paper, we first investigate the difference in the distribution between our Web page visits data and English Wikipedia data. Then on the basis of the difference, we propose Backward PV-DM, which is a modified version of Paragraph Vector. We report the extensive evaluations as well as the details of the improved methods.

Our main contributions are as follows.

- Comparing our Web page visits data with English Wikipedia data, we show the similarity and difference of frequency distributions between the two data. (Section 2.1)
- On the basis of the analysis of our Web page visits data, we propose Backward PV-DM. The difference between the PV-DM and this model is the context window. (Section 4)
- We evaluated our approach using two real-world data sets from an ad network and obtained better results than existing methods. (Section 5)

2. USER ACTIVITIES ON THE WEB

We define A as a set of possible user activities that we consider. For an i-th user u_i, the sequence of activities on the Web is also defined as:

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

where $a_{i,t} \in A$ is a t-th activity of user u_i, and T_i is the size of this sequence.

In this work, we focus on Web page visits and represent each visit $a_{i,t}$ as a URL of the Web page. These URLs are just extracted from logs of Web services. Therefore this

TargetAd 2016: 2nd International Workshop on Ad Targeting at Scale, WSDM, February 22–25, 2016, San Francisco, CA, USA.
.

method of representing the data is easy to use and scalable. Another option is to obtain hashed URLs that users have visited in the past via the data partners in a similar way to the earlier studies [5, 11] for targeting tasks in display advertising. Thus our approach is simple but widely applicable. Since we represent each Web page visit as a URL, we use “Web page visit” and “URL” interchangeably. Our approach can be easily extended to other types of events such as search queries and ad clicks, if available. Therefore, we describe our approach using the generic activities $a_{i,t}$ in Sections 3 and 4.

2.1 Data Analysis on Web Page Visits

In this section, we reveal the difference between our Web page visits data and English Wikipedia data, since we apply an NLP-based approach to our data.

We collect a part of access logs of July 22, 2014 and extract URLs of the Web pages that each user visited. These access logs include one of the mobile apps for smartphones and tablet computers as well as ordinary Web services. Users whose numbers of Web page visits are between 10 and 1000 are sampled. We discard URLs that occurred fewer than five times in the extracted data. If an interval of time between two consecutive page visits exceeds 30 minutes, we consider it as the start of a new session. A session in a sequence of Web page visits corresponds to a sentence in a paragraph or document. Consequently, there are about 3.87 million unique URLs and one billion page visits in the data.

For English Wikipedia data, we preprocess the latest Wikipedia dump using Matt Mahoney’s script[1] and sentence segmenter in NLTK [2].

In summary, we obtained two kind of observations by comparing the data.

- The frequencies of URLs in our Web page visits data follow a power-law distribution. The frequencies of words in English Wikipedia data have the same property, as is widely known [4].
- Focusing on the relative position in a session or sentence, on the other hand, the two distributions of frequencies are significantly different.

The following part describes these two observations in detail.

First, the frequencies of URLs and words in the data are shown in Figure 1. It is widely known that the frequencies of words in most languages follow a power-law distribution [4]. A power-law distribution looks like a roughly straight line of the log-log plot. Clearly, the plot of Web page visits shows an approximately straight line[2]. The exponents of the regression lines with power-law distribution are about -1.0. The plot of the Wikipedia data seems to be a piecewise linear. The exponents of the regression lines are -1.1 for the early part of data and -1.5 for all data. Therefore, the frequencies in both data approximately follow a power law. However, the tail part of Web page visits data is “fatter” than that of English Wikipedia.

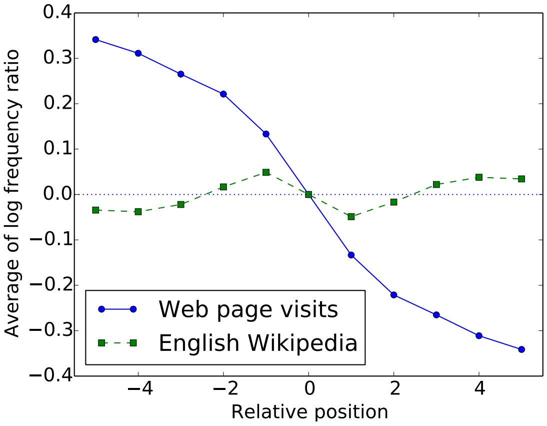

Figure 2: Average of log frequency ratio for relative positions. Because of symmetric property of log ratio (y-axis) and relative position (x-axis), these plots are symmetric with respect to the origin.

Next, the average of log frequency ratio for relative positions are shown in Figure 2. The log frequency ratio for relative position k, that is $a_{i,t}$ and $a_{i,t+k}$, is defined as follows:

$$\log\left(\frac{\mathrm{freq}(a_{i,t+k})}{\mathrm{freq}(a_{i,t})}\right),$$

where $\mathrm{freq}(a_{i,t})$ represents frequency of a Web page visit $a_{i,t}$ (or a word) in the data. We average the log frequency ratio of t and $t+k$ in a session or sentence. The average values of English Wikipedia are around zero, which indicates that word frequencies do not change depending on the position in a sentence. By contrast, the average log frequency ratio of Web page visits data decreases as the relative position k becomes larger. This suggests that URLs that appear in the latter part of a session is the “tail” URLs whereas the URLs that exist in the former part is the “head” URLs. This is caused by a trend of users’ Web browsing behavior. Most users of Yahoo! JAPAN visit the front page [3] at the beginning of the session and then follow the hyperlinks in the Web pages to move to diffferent sites, such as news, sports, finance, and shopping. Similarly, on each site, users visit the Web pages in which they are interested by following the hyperlinks or using the search engine.

According to the above analysis, to capture the users’ interests or preferences suitably, the Web page visits of “tail” URLs that appear in the latter part of the session are more important.

3. EXISTING VECTOR MODELS

In this section, we describe Paragraph Vector [8] and other vector models [10, 6] for our problem settings.

3.1 PV-DM

We first describe the PV-DM, Distributed Memory Model of Paragraph Vectors [8]. The objective of the vector model for an i-th user u_i’s sequence is to maximize the sum of log

[1] `http://mattmahoney.net/dc/textdata.html`

[2] The straight line on the log-log plot is a necessary, but not sufficient, condition for the data following a power-law distribution [4]. Data generated by a log-normal distribution also look roughly straight on the log-log plot.

[3] `http://www.yahoo.co.jp/` for PCs and `http://m.yahoo.co.jp/` for mobile devices.

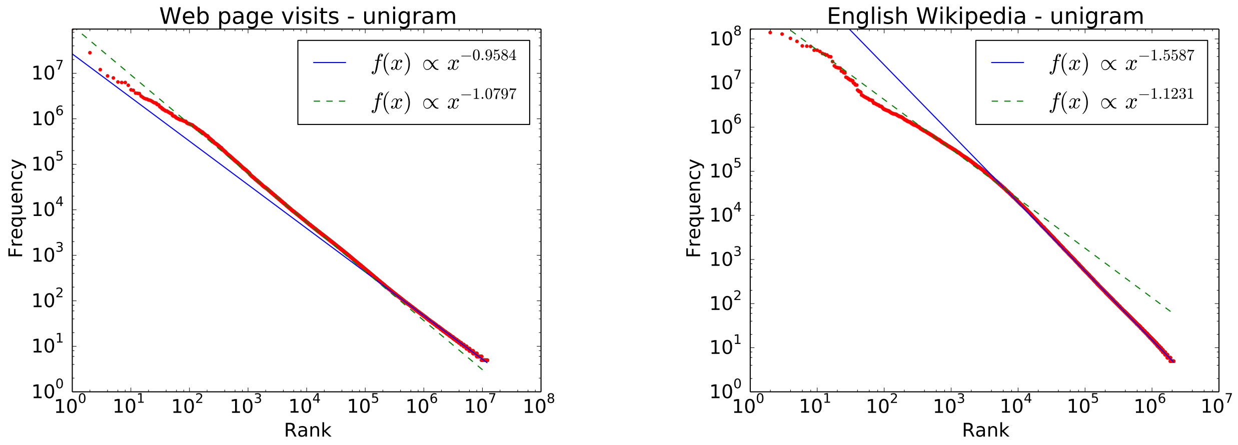

Figure 1: Log-log plot for Web page visits data (Left) and English Wikipedia data (Right). X-axis represents the rank of activity or words in the frequency table, and y-axis is the number of occurrences. The solid and dashed lines represent regression lines for all data and early part of the data (rank less than 10^4), respectively.

probabilities:

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

where s is the size of the context window. This means the conditional probability of the activity $a_{i,t}$ given preceding activities $a_{i,t-1},\, \ldots,\, a_{i,t-s}$ and user u_i. The PV-DM defines the probability of this multi-class problem using the softmax function as follows:

$$p(a_{i,t} \mid a_{i,t-1},\, \ldots,\, a_{i,t-s},\, u_i) := \frac{\exp(\boldsymbol{w}_{a_{i,t}}^{\mathrm{T}} \boldsymbol{v}_I)}{\sum_{a \in A} \exp(\boldsymbol{w}_a^{\mathrm{T}} \boldsymbol{v}_I)}, \qquad (1)$$

where $\boldsymbol{w}_{a_{i,t}}$ is the "output" vector corresponding to $a_{i,t}$ and $\boldsymbol{v}_I$ is the "input" vector corresponding to the previous activities $a_{i,t-1},\, \ldots,\, a_{i,t-s}$ and user u_i. We also define the "input" activity vector corresponding to $a_{i,t}$ as $\boldsymbol{v}_{a_{i,t}}$ and user "input" vector as $\boldsymbol{v}_{u_i}$. Therefore, $\boldsymbol{v}_I$ is represented as a concatenated vector:

$$\boldsymbol{v}_I = [\boldsymbol{v}_{a_{i,t-1}}^{\mathrm{T}},\, \ldots,\, \boldsymbol{v}_{a_{i,t-s}}^{\mathrm{T}},\, \boldsymbol{v}_{u_i}^{\mathrm{T}}]^{\mathrm{T}}.$$

For the case of $j \leq 0$, an input activity vector $\boldsymbol{v}_{a_{i,j}}$ is replaced with a special padding vector $\boldsymbol{v}_{NULL}$. We define the size of input activity vector $|\boldsymbol{v}_{a_{i,j}}|$ as v_a and the size of input user vector $|\boldsymbol{v}_{u_i}|$ as v_u, so the size of both input vector $\boldsymbol{v}_I$ and output vector $\boldsymbol{w}_{a_{i,j}}$ is $s \times v_a + v_u$.

The PV-DM can be regarded as a combination of an abstract word n-gram model and a topic model.

The user vector $\boldsymbol{v}_{u_i}$ is used as a feature vector of various user-related prediction tasks, such as ad click prediction. We also use the "input" activity vectors $\boldsymbol{v}_{a_{i,j}}$ as features and show the effectiveness in the experiment.

3.2 PV-DBoW

The PV-DBoW, Distributed Bag of Words version of Paragraph Vector, is another version of Paragraph Vector [8]. The objective of the PV-DBoW for an i-th user u_i's sequence is to maximize the sum of log probabilities:

$$\sum_t \log p(a_{i,t} \mid u_i).$$

The probability of this multi-class problem is also defined using the softmax function as follows:

$$p(a_{i,t} \mid u_i) := \frac{\exp(\boldsymbol{w}_{a_{i,t}}^{\mathrm{T}} \boldsymbol{v}_{u_i})}{\sum_{a \in A} \exp(\boldsymbol{w}_a^{\mathrm{T}} \boldsymbol{v}_{u_i})}. \qquad (2)$$

For PV-DBoW, the input user vector $v_u = |\boldsymbol{v}_{u_i}|$ and output word vector $\boldsymbol{w}_{a_{i,j}}$ are the same size.

The PV-DBoW can be viewed as a simplified version of PV-DM where the size of the context window s is zero. In other words, this model uses the part of the topic model and omits the part of the abstract word n-gram.

3.3 CBoW and Skip-gram

For comparison with the above Paragraph Vectors, we also describe word vector models, CBoW and Skip-gram model [10].

Similar to Paragraph Vectors, the objective of CBoW and Skip-gram is also to maximize the sum of log probabilities, which is defined using the softmax function. However, these two vector models are proposed for obtaining word representation. Therefore, in our problem settings, these models just provide the representations for activities, not for users directly.

The objective function of the CBoW, Continuous Bag of Words model, is defined as follows:

$$\sum_t \log p(a_{i,t} \mid a_{i,t-s},\ \ldots,\ a_{i,t-1},\ a_{i,t+1},\ \ldots,\ a_{i,t+s}).$$

$$p(a_{i,t} \mid a_{i,t-s},\ \ldots,\ a_{i,t-1},\ a_{i,t+1},\ \ldots,\ a_{i,t+s}) := \frac{\exp(\boldsymbol{w}_{a_{i,t}}^{\mathrm{T}} \boldsymbol{v}_I)}{\displaystyle\sum_{a \in A} \exp(\boldsymbol{w}_a^{\mathrm{T}} \boldsymbol{v}_I)}, \qquad (3)$$

where $\boldsymbol{v}_I$ is the averaged vector of the context vectors:

$$\boldsymbol{v}_I = \frac{1}{2s} \sum_{-s \leq k \leq s, k \neq 0} \boldsymbol{v}_{a_{i,t+k}}.$$

On the other hand, the objective function of the Skip-gram model is as follow:

$$\sum_t \sum_{-s \le k \le s, k \ne 0} \log p(a_{i,t+k} \mid a_{i,t})$$

$$p(a_{i,t+k} \mid a_{i,t}) := \frac{\exp(\boldsymbol{w}_{a_{i,t+k}}^{\mathrm{T}} \boldsymbol{v}_{a_{i,t}})}{\sum_{a \in A} \exp(\boldsymbol{w}_a^{\mathrm{T}} \boldsymbol{v}_{a_{i,t}})}. \qquad (4)$$

The Directed Skip-gram model proposed by Djuric et al. [6] is a modified model that considers the future activities given by the past activity:

$$\sum_t \sum_{0 < k \le s} \log p(a_{i,t+k} \mid a_{i,t}).$$

4. PROPOSED METHOD

In this section, we propose Backward PV-DM. Then, we explain the learning method for these vector models.

4.1 Backward PV-DM

On the basis of the analysis of our Web page visits data in Section 2.1, we propose a modified model named Backward PV-DM. The difference between PV-DM and this model is the context window. The objective of the Backward PV-DM is to maximize the sum of log probabilities:

$$\sum_t \log p(a_{i,t} \mid a_{i,t+1}, \ldots, a_{i,t+s}, u_i).$$

For predicting "output" activity $a_{i,t}$, Backward PV-DM employs the following activities $a_{i,t+1}, \ldots, a_{i,t+s}$ as "input" whereas PV-DM use the previous activities $a_{i,t-1}, \ldots, a_{i,t-s}$. The conditional probability is defined as follows:

$$p(a_{i,t} \mid a_{i,t+1}, \ldots, a_{i,t+s}, u_i) := \frac{\exp(\boldsymbol{w}_{a_{i,t}}^{\mathrm{T}} \boldsymbol{v}_I)}{\sum_{a \in A} \exp(\boldsymbol{w}_a^{\mathrm{T}} \boldsymbol{v}_I)}, \qquad (5)$$

$$\boldsymbol{v}_I = [\boldsymbol{v}_{a_{i,t+1}}^{\mathrm{T}}, \ldots, \boldsymbol{v}_{a_{i,t+s}}^{\mathrm{T}}, \boldsymbol{v}_{u_i}^{\mathrm{T}}]^{\mathrm{T}}.$$

We also present a Reverse PV-DM whose input sequences are just reversed, from future to past. Therefore, the conditional probability of Reverse PV-DM is the same as that of Backward PV-DM, but the sliding directions of the context window are different. The differences between PV-DM, Reverse PV-DM, and Backward PV-DM are summarized in Figure 3.

The sliding direction of the context window does not change the whole objective to be maximized. However, this objective is not concave because of the bilinear form, and we search for a better local maximum of the objective using a stochastic gradient descent (SGD) as described in the following Section 4.2. In our implementation, the sliding direction is the same as the input order of SGD procedure. The user vector in the model acts as a memory that remembers what is missing from the current context. Since the latter input is more memorable than the former input, the sliding direction and input order affect the quality of user vector. In other words, the informative Web page visits that occur in the latter part of a session should be inputted lastly. The experimental results present the effect.

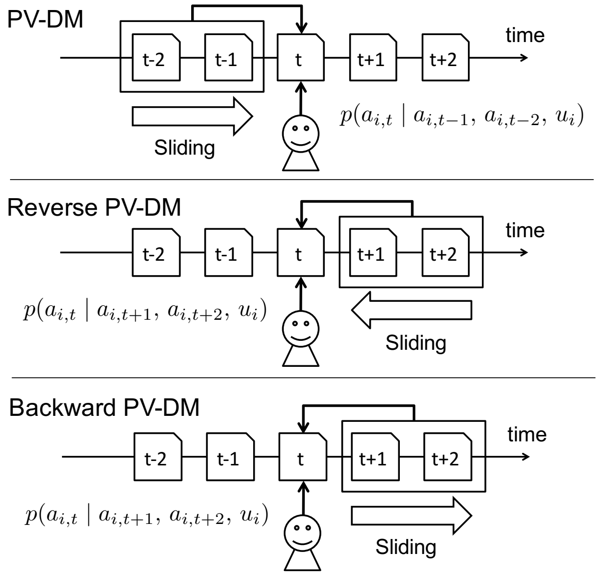
PV-DM
t-2
t-1
t
t+1
t+2
time
Sliding
$p(a_{i,t} \mid a_{i,t-1}, a_{i,t-2}, u_i)$
Reverse PV-DM
t-2
t-1
t
t+1
t+2
time
$p(a_{i,t} \mid a_{i,t+1}, a_{i,t+2}, u_i)$
Sliding
Backward PV-DM
t-2
t-1
t
t+1
t+2
time
$p(a_{i,t} \mid a_{i,t+1}, a_{i,t+2}, u_i)$
Sliding

Figure 3: Illustration of PV-DM, Reverse PV-DM, and Backward PV-DM where the size of the context window s is two. The differences between these models are the conditional probability to be maximized and the sliding direction of context window.

For comparison with Backward PV-DM, we use Backward Skip-gram in the experiment, which is a reversed version of Directed Skip-gram:

$$\sum_t \sum_{-s \le k < 0} \log p(a_{i,t+k} \mid a_{i,t}) = \sum_{t=1}^{T_i} \sum_{0 < k \le s} \log p(a_{i,t-k} \mid a_{i,t}).$$

4.2 Learning the vector models

The computation of Eqs. (1) – (5) and their first derivative is impractical because the number of unique activities $|A|$ is typically large. Le and Mikolov [8] originally used hierarchical softmax with a Huffman binary tree based on word frequencies for fast training. Here, instead of hierarchical softmax, we employ a negative sampling approach [10]. Hence an alternate objective to $\log p(a_{i,t} \mid a_{i,t-1}, \ldots, a_{i,t-s}, u_i)$ with Eq. (2) is defined as:

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

where $\sigma(z) = 1/(1+\exp(-z))$ is a sigmoid function, k is the number of randomly sampled negative instances, and $p_n(a)$ is a noise distribution generating negative instances. We use the "unigram" distribution $U(a)$ raised to the 3/4th power as $p_n(a)$ in the same way as Mikolov et al. [10] did. We train the model using asynchronous SGD [13] with AdaGrad [7]. In the inference step for new users, the user vectors $\boldsymbol{v}_u$ are learned while input and output activity vectors $\boldsymbol{v}_a$ and $\boldsymbol{w}_a$ are fixed.

5. EXPERIMENTS

In this section, we evaluated our approach using two real-world data sets from Web services of Yahoo! JAPAN.

5.1 Data sets

We evaluated the proposed method using two supervised learning data sets: *AdClicker* and *SiteVisitor*. *AdClicker*

Table 1: Statistics for two data sets. #Features is the number of unique URLs that occurred more than or equal to five times in each data set.

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

consists of the users who clicked contextual ads that are included in the five selected ad campaigns. Similarly, *SiteVisitor* consists of the users who visited Web sites of five selected advertisers.

For simplicity, we created these two data sets in view of predicting a user's particular activities on a day on the basis of the history of Web pages visited the previous day. The training and validation sets were generated from logs of July 22 and 23, 2014. Web page visits on the former day are used as features, and the target activity in the latter day is treated as labels. Similarly, a test set was generated from logs of July 23 and 24, 2014, as features and labels, respectively. Since these features were extracted from Web service logs of Yahoo! JAPAN, they are only a small fraction of the entire user activities on the Web. These features do not include visits to advertisers' sites, which are the labels of *SiteVisitor*.

Contextual ads in *AdClicker* are determined to be displayed by the Web page content as well as user information. Therefore, learning each Web page representation is also helpful for this task. On the other hand, *SiteVisitor* is the data set based on more complicated user interests.

The statistics for data sets are summarized in Table 1.

5.2 Evaluation settings

AdClicker and *SiteVisitor* are multi-label data sets because a user can click more than one ad or visit various advertisers' sites. In the experiment, we transformed the multi-label problem into a set of binary classification problems. We represent the binary classification tasks for *AdClicker* as Ac1 to Ac5 and *SiteVisitor* as Sv1 to Sv5. For each binary classification task, we trained logistic regression classifiers using features extracted by each method. The evaluation measure is Area Under ROC Curve (AUC).

5.3 Proposed methods and baselines

We compared the methods using Paragraph Vector with some baselines. *Bin* and *Freq* are weak baselines that use raw URLs as features. *Freq* takes into account the frequencies of the user's site visits, whereas *Bin* considers only whether a user visits the Web page or not. Feature vectors of these two methods are high dimensional sparse vectors.

We refer to CBoW, Skip-gram, Directed Skip-gram, and Backward Skip-gram as word vector models. We also refer to PV-DM, Reverse PV-DM, Backward PV-DM, and PV-DBoW as Paragraph Vectors. By using the word vectors models, a user is represented as the simple averaging of input activity vectors $\boldsymbol{v}_a$ in the sequence, which is similar to the approach of Djuric et al. [6]. We use the user vectors $\boldsymbol{v}_u$ in the Paragraph Vectors as user representations. These methods using the vector models are represented in italic form. For example, the proposed method using the PV-DM model is represented as *PV-DM*.

For PV-DM and Backward PV-DM, we also use the averaging of input activity vectors $\boldsymbol{v}_a$ in the same way as the word vector models. We concatenated the user vectors and the averaged vector for the input of prediction tasks. These methods are called *PV-DM(both)* and *Backward PV-DM(both)*. In addition, we evaluated a method that uses the concatenated vectors learned by the PV-DM and Skip-gram model. This method is called *PV-DM+Skip-gram*.

The settings of learning the vector models are as follows: the size of input vectors $v_a = v_u = 400$, the size of context window $s = 5$, the number of randomly sampled negative instances $k = 5$, and the number of epochs (full pass through the data) is five. For Paragraph Vectors, we create the user vectors $\boldsymbol{v}_u$ via an inference step, considering the all users as new users. Because of stochastic behavior of asynchronous SGD and random initialization, we report the mean value of five runnings for the methods using vector models.

5.4 Results

The experimental results are summarized in Table 2. The **bold** elements indicate the best performance of the methods. The underlined scores are the best results of the word vector models and Paragraph Vectors.

As reported in our work-in-progress paper [15], *PV-DM* achieved better results than *Skip-gram* in *SiteVisitor* whereas the opposite trend is shown in *AdClicker*. This is caused by the difference between two data sets as described above. Two weak baselines *Bin* and *Freq*, which use raw URLs as features, perform poorly for almost all cases.

Backward PV-DM achieved better results than *PV-DM* and *Reverse PV-DM* consistently. As described in Section 4.1, the objectives of *Backward PV-DM* and *Reverse PV-DM* are the same. The difference between these models is just the sliding direction of context window, in other words, the input order of SGD procedure. However, since the Web page visits that appear in the latter part of a session have more information of the user's interests, the direction and input order are important to improve the quality of user vectors, which can act as a memory of the interests. On the other hand, the results of *Backward Skip-gram* are not as good as those of *Skip-gram* and *Directed Skip-gram*.

Backward PV-DM(both) achieved the best results in seven of ten tasks. *PV-DM(both)* is also better than *PV-DM*. These results show the effectiveness of an approach that uses the averaging of input activity vectors as well as the user vector learned by PV-DM and Backward PV-DM.

6. RELATED WORK

In the online advertising field, some previous works focused on finding the user segments that might be interested in a given advertiser's products, inferred from web-browsing behavior information. These approaches are known as behavioral targeting [1] or conversion optimization [9, 11]. Advertisers increase the effectiveness of advertising to deliver their ads to the audience found by the approaches.

Perlich et al. [11] presented a transfer learning approach for online display targeting. In the first stage of the approach, users are represented as a bag-of-words representation of the users browsing history, with each URL hashed into its own binary feature.

Djuric et al. [6] proposed an approach for improving estimation of ad click or conversion probability on the basis of a sequence of a user's online actions modeled using the Hidden Conditional Random Fields (HCRF) model [12]. To address the sparsity issue at the input side of the HCRF model, the authors proposed a directed version of the Skip-gram model,

Table 2: Experimental results. Values are *AUC*. We report the mean value of five runnings for the methods using vector models (see Section 5.3 for more details).

	AdClicker					*SiteVisitor*				
	Ac1	Ac2	Ac3	Ac4	Ac5	Sv1	Sv2	Sv3	Sv4	Sv5
Bin	0.9753	0.8063	**0.6641**	0.7052	0.7524	0.7619	0.8188	0.7087	0.7920	0.7292
Freq	0.9814	0.8184	0.6580	0.6961	0.7509	0.7821	0.8163	0.7006	0.7781	0.7256
CBoW	0.9903	0.8323	0.6533	0.7154	0.7700	0.7999	0.8277	0.7067	0.7849	0.7339
Skip-gram	0.9906	0.8354	0.6562	0.7163	0.7725	0.8017	0.8328	0.7135	0.7931	0.7417
Directed Skip-gram	0.9904	**0.8374**	0.6533	0.7159	0.7706	0.8019	0.8308	0.7120	0.7914	0.7394
Backward Skip-gram	0.9905	0.8328	0.6525	0.7138	0.7712	0.8018	0.8307	0.7125	0.7909	0.7388
PV-DM	0.9899	0.8151	0.6483	0.7242	0.7633	0.8051	0.8343	0.7180	0.7964	0.7479
Reverse PV-DM	0.9884	0.8263	0.6481	0.7274	0.7618	0.8015	0.8345	0.7207	0.7990	0.7489
Backward PV-DM	0.9902	0.8247	0.6537	0.7345	0.7661	0.8092	0.8366	0.7222	0.8028	0.7491
PV-DBoW	0.9894	0.8288	0.6507	0.7290	0.7581	0.7965	0.8294	0.7198	0.7945	0.7489
PV-DM(both)	0.9910	0.8193	0.6531	0.7379	0.7704	0.8134	0.8373	0.7229	0.7998	0.7506
Backward PV-DM(both)	**0.9914**	0.8281	0.6575	**0.7463**	**0.7760**	**0.8162**	**0.8396**	**0.7276**	**0.8069**	0.7513
PV-DM+Skip-gram	0.9912	0.8358	0.6622	0.7391	0.7752	0.8128	0.8395	0.7254	0.8023	**0.7529**

which maximizes log-probabilities of future activities given their preceding activity. Input “words” of the Directed Skip-gram model consist of entities found on a Web page visited by the user and tokens in search queries.

Another line of research attempts to predict the CTR of ads. Predictions of CTR for ads are generally based on a statistical model trained by using past click data. The accuracy of the model depends greatly on the design of the features. Some user-related features were also proposed [3]. Zhang et al. [16] proposed a framework based on Recurrent Neural Networks (RNN) for click prediction of sponsored search advertising. This framework directly models the dependency on a user’s sequential behaviors into the click prediction process through the recurrent structure in RNN.

For an English to French translation task, Sutskever et al. [14] reported that the reversed input of the words in the source sentence to a Long Short-Term Memory (LSTM) model achieved the better result. This technique is related to our discussion of the difference between Reverse PV-DM and Backward PV-DM.

7. CONCLUSION AND FUTURE WORK

In this paper, on the basis of the analysis of our Web page visits data, we proposed Backward PV-DM, which is a modified version of Paragraph Vector. We evaluated this approach on two ad-related data sets based on logs from Web services of Yahoo! JAPAN. Experimental results demonstrated the effectiveness of our proposed method.

Our future work will take three directions. First, we want to study the use of various types of features such as search queries and Web page contents, while we focus on the URLs of Web pages in this paper, for simplicity and scalability. Second, we plan to investigate a method that obtains user representations using learning other than unsupervised learning, such as semi-supervised, multi-label, and multi-task learning. Finally, we are also interested in the sequence modeling with LSTM RNNs and efficient learning methods for Web scale user data.

8. REFERENCES

[1] M. Aly, A. Hatch, V. Josifovski, and V. K. Narayanan. Web-scale user modeling for targeting. In *WWW Companion*, 2012.

[2] S. Bird, E. Klein, and E. Loper. *Natural Language Processing with Python.* O’Reilly Media, Inc., 2009.

[3] H. Cheng and E. Cantú-Paz. Personalized click prediction in sponsored search. In *WSDM*, 2010.

[4] A. Clauset, C. R. Shalizi, and M. E. J. Newman. Power-law distributions in empirical data. *SIAM Review*, 2009.

[5] B. Dalessandro, D. Chen, T. Raeder, C. Perlich, M. Han Williams, and F. Provost. Scalable hands-free transfer learning for online advertising. In *KDD*, 2014.

[6] N. Djuric, V. Radosavljevic, M. Grbovic, and N. Bhamidipati. Hidden conditional random fields with distributed user embeddings for ad targeting. In *ICDM*, 2014.

[7] J. Duchi, E. Hazan, and Y. Singer. Adaptive subgradient methods for online learning and stochastic optimization. *J. Mach. Learn. Res.*, 2011.

[8] Q. Le and T. Mikolov. Distributed representations of sentences and documents. In *ICML*, 2014.

[9] Y. Liu, S. Pandey, D. Agarwal, and V. Josifovski. Finding the right consumer: Optimizing for conversion in display advertising campaigns. In *WSDM*, 2012.

[10] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *NIPS*, 2013.

[11] C. Perlich, B. Dalessandro, T. Raeder, O. Stitelman, and F. Provost. Machine learning for targeted display advertising: transfer learning in action. *Machine Learning*, 2014.

[12] A. Quattoni, S. Wang, L. Morency, M. Collins, and T. Darrell. Hidden conditional random fields. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 2007.

[13] B. Recht, C. Re, S. Wright, and F. Niu. Hogwild: A lock-free approach to parallelizing stochastic gradient descent. In *NIPS*, 2011.

[14] I. Sutskever, O. Vinyals, and Q. V. V. Le. Sequence to sequence learning with neural networks. In *NIPS*, 2014.

[15] Y. Tagami, H. Kobayashi, S. Ono, and A. Tajima. Modeling user activities on the web using paragraph vector. In *WWW Companion*, 2015.

[16] Y. Zhang, H. Dai, C. Xu, J. Feng, T. Wang, J. Bian, B. Wang, and T.-Y. Liu. Sequential click prediction for sponsored search with recurrent neural networks. In *AAAI*, 2014.