Recurrent Neural Networks for Customer Purchase Prediction on Twitter

Mandy Korpusik
Computer Science & Artificial Intelligence Laboratory, MIT
Cambridge, Massachusetts
korpusik@mit.edu

Shigeyuki Sakaki
Fuji Xerox Co., Ltd., Japan
sakaki.shigeyuki@fujixerox.co.jp

Francine Chen Yan-Ying Chen
FX Palo Alto Laboratory, Inc.
Palo Alto, California
{chen, yanying}@fxpal.com

ABSTRACT

The abundance of data posted to Twitter enables companies to extract useful information, such as Twitter users who are dissatisfied with a product. We endeavor to determine which Twitter users are potential customers for companies and would be receptive to product recommendations through the language they use in tweets after mentioning a product of interest. With Twitter's API, we collected tweets from users who tweeted about mobile devices or cameras. An expert annotator determined whether each tweet was relevant to customer purchase behavior and whether a user, based on their tweets, eventually bought the product. For the relevance task, among four models, a feed-forward neural network yielded the best cross-validation accuracy of over 80% per product. For customer purchase prediction of a product, we observed improved performance with the use of sequential input of tweets to recurrent models, with an LSTM model being best; we also observed the use of relevance predictions in our model to be more effective with less powerful RNNs and on more difficult tasks.

CCS Concepts

•**Information systems** → **Social recommendation; Personalization;** Social networks;

Keywords

Deep learning, Recommender systems, Microblogs

1. INTRODUCTION

In social media, popular aspects of customer relationship management (CRM) include interacting with and responding to individual customers and analyzing the data for trends and business intelligence. Another mostly untapped aspect is to predict which users will purchase a product, which is useful for recommender systems when determining which users will be receptive to specific product recommendations. Many people ask for input from their friends before buying a higher-priced product, and users will often turn to social media for that input [13]. Many users also post announcements about significant purchases. Thus, social media posts may contain cues that can be used to identify users who are likely to purchase a product of interest as well as later post(s) indicating that a user made a purchase.

Looking to buy a camera for X-mas, can I get some help?
Mann wish I had an iphone,I wanna be cooolll too! #sadtweet Now thinking of getting a new phone
My baby arrived #panasonic #lumix - its waterproof but I daren't try it hah :)
Got a #Xoom tablet today. Already rooted and master of my domain. Thanks @koush

Table 1: Sample tweets indicating a user wants to buy (top three) and bought (bottom two) a product.

In this paper, we present our investigations of deep learning methods for predicting likely buyers from microblogs, specifically Twitter tweets. While many social media posts are private, semi-private or transient, and thus hard to obtain, microblogs such as tweets are generally public, simplifying their collection. With the ability to identify likely buyers, as opposed to targeting anyone who mentions a product, advertisements and products can be presented to a more receptive set of users while annoying fewer users with spam.

Microblogs cover a variety of genres, including informative, topical, emotional, or "chatter." Many of a user's tweets are not relevant, that is, indicative of whether a user is likely to purchase a product. Thus, we hypothesize that identifying tweets that are relevant to purchasing a product is useful when predicting whether a user will purchase that product. We also hypothesize that a sequence of tweets contains embedded information that can lead to more robust prediction than classification of individual tweets. For example, with the given order of the second and third tweets in Table 1, the user seems interested in buying a new phone, but if the order is reversed, the inference is that the user was thinking of buying a phone, but did not buy one. In this work, we

- investigate the use of recurrent neural networks to model sequential information in a user's tweets for purchase behavior prediction. Our use of recurrent models enables previous tweets to serve as context.
- introduce relevance prediction into the model for reducing the influence from noisy tweets.

This paper is organized as follows. In the next section, we describe related work in deep learning and social media product interest prediction. Afterward, we detail our data

CBRecSys 2016, September 16, 2016, Boston, MA, USA.

Tweet Type	Regular Expression
bought	"my new .* X", "gotta new .* X", "bought a .* X", "splurged on a .* X"
want	"should I buy .* X", "wanna .* X", "should I go for .* X", "need a new .* X"

Table 2: Sample of expressions used to identify candidate bought/want users, where X is a product.

Final User Label	Candidate Want User	Candidate Buy User
Buy User	64	2491
Not-Buy User	1226	315

Table 3: Corpus statistics of annotated candidate users collected via tweets containing want/buy expressions that were then labeled buy/not-buy.

collection and annotation process. Then, we explain the deep learning methods, discuss the experiments, and analyze results. Finally, we conclude and propose future work.

2. RELATED WORK

There are several works related to identifying customers with interest in a product. A system developed by [14] for predicting user interest domains was based on features derived from Facebook profiles to predict from which category of products an eBay user is likely to make a purchase. [9] used a rule-based approach for identifying sentences indicating "buy wishes" from forums with buy-sell sections. [3] presented a method for predicting whether a single post on Quora or Yahoo! Answers contained an expression of "purchase intent." However, some of their Purchase Action words, including "want" and "wish" only indicate interest in a product. Although users may say they want a product, many of these Twitter users will not buy the product in the near future (see Figure 2 and Table 3). For our task, we predict whether a user will actually make a purchase; this is different from the task of predicting user interest in a product. Often users with interest in a product may not have the means to purchase soon or may say they want or need something as an indication that they like something.

Our approach also differs from these works in that access to Facebook profiles is not needed, and features are automatically learned using neural networks. In addition, most of the earlier works classify a single sentence or posting. In contrast, we predict user behavior based on past postings, that is, from a *sequence of tweets*, which enables preceding tweets in a sequence to provide context for the current tweet.

Our approach draws on the deep learning Recurrent Neural Network (RNN) models which have been successfully applied to sequential data such as sentences or speech. For example, [5] used a combination of a convolutional neural network to represent sentences and an RNN to model sentences in a discourse. [2] observed that long short-term memory (LSTM) models, a type of RNN with longer memory, performed better than an RNN on the TIMIT phoneme recognition task. In our work, we compare RNN and LSTM models for the task of predicting whether a user will buy a product of the type that they have mentioned.

3. DATA COLLECTION AND LABELING

For the buy prediction task, we focused on two product categories: (1) cameras and (2) mobile devices, i.e., mobile phones, tablets, and smart watches. These are generally higher-priced products which users do not purchase frequently and therefore are more likely to tweet about.

We created a separate corpus for each category composed of tweets by users who either: (1) bought a target product or (2) wanted, but did not buy, a target product [10]. To collect tweets by each user, we first identified from eBay listings a set of model names for each product category. Similar model names were merged, e.g., "iPhone4" and "iPhone5" into "iPhone," resulting in 146 camera names and 80 mobile device names. We also created a set of regular expressions that may indicate a user bought or wanted one of the products (see sample in Table 2). Tweets containing a bought or want expression for one of the product names were then collected using the Twitter search API, and the user of each tweet was identified from the tweet meta-data. The tweets of the identified users were collected using the Twitter search and timeline APIs. We called users found with "bought" regular expressions *candidate buy users*, and users identified from "want" regular expressions *candidate want users*.

Due to poor labeling performance by Mechanical Turkers, who often were not familiar with many of the lesser-known mobile devices and cameras, we used an "expert" annotator to whom we gave many examples of labeled bought and want tweets, including trickier cases. For example, a user did not buy a camera if they were given one or if they retweeted (RT) a user who bought a camera. A sample of the labels was checked for accuracy by one of the authors. The annotator determined whether each *candidate want user* tweeted that they bought the product type of interest; if so, the *candidate want user* was labeled a *buy user* (Table 3). Similarly, the tweets of each *candidate buy user* were examined for at least one tweet indicating that the user really bought the target product type. In total, we annotated tweets from 2,403 mobile device users and 1,252 camera users. The annotator also labeled a separate random sample of tweets as relevant/not to predicting whether a target product was bought.

4. DEEP LEARNING METHODS

In this section, we describe the neural network (NN) models we implemented in Python's Theano toolkit [1] for classifying tweets as relevant/not and predicting whether a Twitter user bought a product 60 days after tweeting about it.

The logistic regression (**LR**) model combines the input with a weight matrix and bias vector, feeding it through a softmax classification layer that yields probabilities for each class i. The class i with the highest probability is the output.

A feed-forward (**FF**) network enables more complex functions to be computed through the addition of a sigmoid hidden layer below the softmax. A natural extension of the **FF** network for sequences is a recurrent neural network (**RNN**), in which the hidden layer from the previous timestep is fed into the current timestep's hidden layer:

$$h_t = \sigma(W_x x_t + W_h h_{t-1} + b) \quad (1)$$

where h_t is the hidden state, x_t is the input vector, W is a learned weight matrix, and b is a learned bias vector.

Thus, information from early words/tweets is preserved across time and is still accessible upon reaching the final word/tweet for making a prediction. However, typically the error gradient vanishes as the sequence becomes increasingly long, which in practice causes information loss over long time

spans. To compensate for this, long short-term memory [4] uses input, output, and forget gates to control what information is stored or forgotten within a memory cell over longer periods of time than a standard RNN. In the **LSTM**, the input gate i_t, forget gate f_t, and candidate memory cell $\widetilde{C_t}$ are computed using input x_t and previous hidden layer h_{t-1}, weight matrices (i.e., W_i and U_i for the input gate, W_f and U_f for the forget gate, and W_c and U_c for candidate $\widetilde{C_t}$), and forget gate bias term b_f as follows:

$$i_t = \sigma(W_i x_t + U_i h_{t-1}) \quad (2)$$

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f) \quad (3)$$

$$\widetilde{C_t} = \tanh(W_c x_t + U_c h_{t-1}) \quad (4)$$

The new memory cell C_t is generated from the combination of the candidate memory cell $\widetilde{C_t}$, controlled by the input gate i_t through element-wise multiplication, and the previous memory cell C_{t-1}, modulated by the forget gate f_t:

$$C_t = i_t * \widetilde{C_t} + f_t * C_{t-1} \quad (5)$$

The output gate and new hidden layer are computed by:

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + V_o C_t) \quad (6)$$

$$h_t = o_t * \tanh C_t \quad (7)$$

To preprocess the data, each tweet was first tokenized by the TweeboParser [6], a dependency parser trained on tweets. Then each token was converted into a vector representation using word2vec [7], which learns an embedded vector representation from the weights of a neural network trained on Google News. We also experimented with the GloVe word embedding [8] and with learning an embedding from random initialization. In preliminary experiments on predicting tweet relevance, word2vec consistently performed best and so was used for all reported results.

4.1 Models for Predicting Tweet Relevance

To predict tweet relevance, we compared the LR, FF, RNN, and LSTM models. Since the RNN and LSTM are sequential, the input was an array of token embedded vectors; for the LR model and FF networks, we summed the token embedded vectors as input. For regularization of the LR and FF models, we employed an early stopping technique, and for the RNN and LSTM networks, we incorporated dropout.

4.2 Models for Predicting Purchase Behavior

To predict whether a user will buy a product based on their tweets, we propose a configuration of neural networks that uses predicted tweet relevance in purchase prediction. The input for each user is a sequence of *tweets* (instead of words, as is more commonly used) enabling the preceding tweets to provide context for the current tweet. To model the information in a tweet sequence, a recurrent network (e.g., RNN or LSTM) is intuitively a good choice.

In our proposed joint model (Figure 1), tweets from a user are input as a sequence where each tweet is represented as the sum of the embedded vectors representing its words. The (optional) lower sub-network predicts the relevance of each tweet; we use the best of the four types of relevance classification models, the feed-forward neural net.

The buy sub-network at each time step (i.e., tweet) is composed of either an RNN or an LSTM memory cell, which is fed a tweet vector along with its predicted relevance. The maximum across each dimension of all the tweets' hidden layer outputs are fed into the softmax classifier for the final buy/not buy prediction. The softmax will generalize well to future work on predicting other labels besides buy/not-buy.

For all experiments, each data set was split into 10 partitions for 10-fold cross-validation. Within each fold, 10% was for validation, 10% for testing, and the rest for training. In order to incorporate the FF classifier for predicting tweet relevance as a sub-network in a joint network, we trained a separate classifier for each of the 10 cross-validation partitions. We selected all the users in the training and validation sets for that partition, and trained the relevance classifier on all those users' tweets for which we had a relevance label. The predicted relevance for each tweet was used as an additional feature to predict the user's purchase behavior.

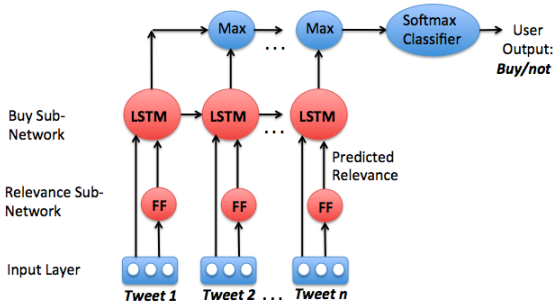

Figure 1: Purchase behavior prediction network.

5. EXPERIMENTS AND RESULTS

We conducted two sets of experiments: first, predicting whether a tweet is relevant to a user's purchase behavior; second, predicting whether a Twitter user will eventually purchase a product within 60 days of tweeting about it.

For all the networks, we set the hidden layer size to 50. The weight matrices were initialized randomly, and bias vectors were initialized to zero. We used RMSprop [12] and negative log-likelihood for training, sigmoid nonlinear activations, and batches of size 10 for up to 100 epochs.

5.1 Tweet Relevance

We observe from the results shown in Table 4 that the FF model performed best. The task is harder for combined mobile device and camera data than for the two individual products, likely due to differences between domain-specific relevance indicators for cameras versus mobile devices.

Model	Mobile	Camera	Both
Logistic	79.7	78.8	74.7
FF	**81.2**	**80.4**	**78.0**
RNN (25%)	80.1	79.2	77.7
LSTM (50%)	80.2	77.0	77.0

Table 4: Accuracy of learning models for tweet relevance. The best dropout rates are 25% and 50% [11].

5.2 User Purchase Behavior

We explored RNNs and LSTMs for predicting whether or not a Twitter user would buy a product, since these models sequentially scan through a user's tweets. We incorporated the FF tweet relevance prediction model as a sub-network in

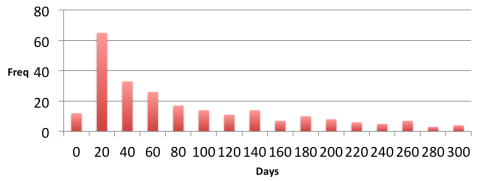

Figure 2: Histogram of the number of days between the first "want" and "bought" tweets by "buy" users.

a deep network; this model predicts a feature indicating each tweet's relevance, which is appended to the input tweets.

For each Twitter user, we used all tweets containing a product mention within a 60-day span, limited to users who wrote between five and 100 product-related tweets; the upper limit was used to filter out advertisers. The 60 days are motivated by the assumption that companies are not interested in promoting products for longer periods, and by Figure 2, which shows that about half the users purchase what they want within 60 days.

We evaluated our models on mobile devices, cameras, and the two combined. Negative examples included Twitter users who wanted a product but did not mention buying it. We trained on their tweets from within the 60-day window before their most recent tweet that mentions wanting a product. Positive examples included users who eventually tweeted about buying a product (the *buy users*), but did not include the "bought" tweet or any tweets written afterward.

The 10-fold cross-validation results averaged over several runs (due to random initialization) on the expert-labeled training data are shown in Table 5. As the baseline, we trained a FF model with sums of all tokens across all tweets for each user (*-sum) as the input. We observe better performance when tweet information is input sequentially to a model with memory (*-seq) than when the sequence information is lost by summing (*-sum) tweets. That is, it is important to capture information embedded in the sequence of tweets. As expected, the LSTM consistently outperformed the RNN because it has the ability to retain information over longer time spans. We also observe that the addition of predicted relevance probabilities to the RNN model (*+Rel) improved performance over the simple RNN; however, adding tweets' relevance only improved the LSTM's performance for the harder combined product task, which may indicate that the vanilla LSTM is powerful enough to learn which tweets are relevant or not when trained on a single product.

Model	Mobile	Camera	Both
FF-sum	73.6	66.3	73.4
RNN-seq	80.8	78.0	79.3
RNN-seq+Rel	81.3	80.5	80.1
LSTM-seq	**83.9**	**81.4**	81.5
LSTM-seq+Rel	83.8	80.9	**81.7**

Table 5: Purchase prediction cross-validation.

6. CONCLUSION

In this work, we investigated deep learning techniques for predicting customer purchase behavior from Twitter data that recommender systems could leverage. We collected a labeled corpus of buy/not buy users and their tweets.

A FF neural network performed best at predicting whether a tweet is relevant to purchase behavior, with an accuracy of 81.2% on mobiles devices and 80.4% on cameras. We found that the use of a deep learning model that incorporates sequential information performed better than ignoring sequential information for the purchase prediction task.

Our initial work in this area has many possible extensions. While we used a 60-day window, it would be interesting to observe user purchase probability changes over time. We will also predict related behaviors, e.g., product comparison.

7. REFERENCES

[1] F. Bastien, P. Lamblin, R. Pascanu, J. Bergstra, I. J. Goodfellow, A. Bergeron, N. Bouchard, and Y. Bengio. Theano: New features and speed improvements. Deep Learning and Unsupervised Feature Learning NIPS 2012 Workshop, 2012.

[2] A. Graves, A.-r. Mohamed, and G. Hinton. Speech recognition with deep recurrent neural networks. In *Proc. of ICASSP*, pages 6645–6649. IEEE, 2013.

[3] V. Gupta, D. Varshney, H. Jhamtani, D. Kedia, and S. Karwa. Identifying purchase intent from social posts. In *Proc. of ICWSM*. AAAI, 2015.

[4] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

[5] N. Kalchbrenner and P. Blunsom. Recurrent convolutional neural networks for discourse compositionality. In *Proc. of the 2013 Workshop on Continuous Vector Space Models and their Compositionality*. ACL, 2013.

[6] L. Kong, N. Schneider, S. Swayamdipta, A. Bhatia, C. Dyer, and N. A. Smith. A dependency parser for tweets. In *Proc. of EMNLP*. ACL, 2014.

[7] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *Proc. of NIPS*, pages 3111–3119, 2013.

[8] J. Pennington, R. Socher, and C. Manning. GloVe: Global vectors for word representation. In *Proc. of EMNLP*. ACL, 2014.

[9] J. Ramanand, K. Bhavsar, and N. Pedanekar. Wishful thinking finding suggestions and 'buy' wishes from product reviews. In *Workshop on Computational Approaches to Analysis and Generation of Emotion in Text*, page 54, 2010.

[10] S. Sakaki, F. Chen, M. Korpusik, and Y. Chen. Corpus for customer purchase behavior prediction in social media. *LREC*, 2016.

[11] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

[12] T. Tieleman and G. Hinton. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural Networks for Machine Learning*, 2012.

[13] X. Wang, C. Yu, and Y. Wei. Social media peer communication and impacts on purchase intentions: A consumer socialization framework. *Journal of Interactive Marketing*, 26(4):198–208, 2012.

[14] Y. Zhang and M. Pennacchiotti. Predicting purchase behaviors from social media. In *Proc. of WWW*, pages 1521–1532. ACM, 2013.