Ranking Convolutional Recurrent Neural Networks for Purchase Stage Identification on Imbalanced Twitter Data

Heike Adel[1*], **Francine Chen**[2] **and Yan-Ying Chen**[2]

[1]Center for Information and Language Processing (CIS), LMU Munich, Germany
[2]FX Palo Alto Laboratory, Palo Alto, California, USA
heike@cis.lmu.de
{chen|yanying}@fxpal.com

Abstract

Users often use social media to share their interest in products. We propose to identify purchase stages from Twitter data following the AIDA model (Awareness, Interest, Desire, Action). In particular, we define the task of classifying the purchase stage of each tweet in a user's tweet sequence. We introduce RCRNN, a Ranking Convolutional Recurrent Neural Network which computes tweet representations using convolution over word embeddings and models a tweet sequence with gated recurrent units. Also, we consider various methods to cope with the imbalanced label distribution in our data and show that a ranking layer outperforms class weights.

1 Introduction

As the use of social media grows, more users are sharing interests or experiences with products, and asking friends for information (Morris et al., 2010). Thus, social media posts can contain information useful for marketing and customer relationship management, including user behavior, opinions, and purchase interest.

In this paper, we present a ranking-based, deep learning approach to automatically identify stages in a sales process following the well-known AIDA (Awareness/Attention, Interest, Desire, and Action) model (Lewis, 1903; Dukesmith, 1904; Russell, 1921). Since we are interested in purchases, we define "Action" as buying a product. Knowledge of a user's purchase stage can help to personalize the type of advertisement a user is shown, e.g., while a user with interest may be shown information about product features by a manufacturer, a user with the desire to purchase may be given coupons for a particular store offering the product of interest. In addition to automatically recognizing the traditional AIDA stages, we also add a class with negative sentiment, namely unhappiness of a user with a product.

Attention (A)	i seem to always be debating another iphone
Interest (I)	Should I pre-order a Lumia 650 ?
	I want a lowish end phone , but the 650 looks SO much nicer than the 550
Desire (D)	So i guess it's time to get an iPhone
Bought (B)	JUST GOT THE NEW IPHONE 3s !!! #textme #popular
Unhappiness (U)	I hate my phone
No PS (N)	Who else has an Apple Watch ?
	Learned I can draw you little pictures & notes from my watch

Table 1: Example tweets for the different purchase stages (PS)

Given a user's tweet sequence, we define the purchase stage identification task as automatically determining for each tweet whether the user expresses interest in, wants to buy, or has recently bought a product, etc. Table 1 shows one randomly picked example for each of the purchase stages as well as for an artificial class 'N' which we use for tweets not expressing a purchase stage.

We introduce RCRNN (ranking convolutional recurrent neural network), a hierarchical neural network that uses convolution to create a tweet representation and recurrent hidden layers to represent a tweet sequence. We compare RCRNN with other possible neural network (NN) architectures and non-neural models.

A particular challenge of our dataset is class imbalance: There are much more tweets expressing none of the purchase stages than tweets expressing one of them. We investigate the use of a ranking layer in our NN and compare it against class weights for handling imbalanced data.

[*]The work was performed during an internship at FX Palo Alto Laboratory

To sum up, our contributions are as follows: (1) We define the new task of purchase stage identification from tweets. Our results show that tweets do contain signals indicative of purchase stages. (2) We propose RCRNN, a hierarchical deep learning model to represent tweets and tweet sequences. (3) We show that a ranking layer approach outperforms commonly used class weights for training neural networks on imbalanced data.

2 Related Work

An increasing amount of research is focused on social media with various classification goals. For example, Twitter tweets have been used for the prediction of movie revenues (Asur and Huberman, 2010) and stock prices (Kharratzadeh and Coates, 2012; Bollen and Mao, 2011). Lassen et al. (2014) predicted quarterly iPhone sales motivated by the AIDA model, but did not model AIDA directly as we do in this paper.

More related to our task is classifying whether a user has purchase intent. Vieira (2015) and Lo et al. (2016) used features from e-commerce or content discovery platforms to predict buying intentions. Manually crafted linguistic and/or statistical features have been used to predict potential purchase intent from Quora and Yahoo! Answers (Gupta et al., 2014), and to detect purchase intent in product reviews (Ramanand et al., 2010). The task of identifying purchase intent is related to our task of identifying purchase stages, but does not indicate a users stage in making a purchase decision. The posts in both Quora and Yahoo! Answers, by their nature, tend to be posts by people seeking information, of which some are related to purchase decisions. And the product reviews in Ramanand et al. (2010) are more targeted towards the product being reviewed. All three types of data tend to be less noisy than a user's tweets due in part to a smaller proportion of tangential text, such as "My brother hid my phone".

Works which use Twitter tweets as input largely employ manually-crafted linguistic and statistical features. Hollerit et al. (2013) trained different classifiers on the words and part-of-speech tags of tweets to detect whether a tweet contained "commercial intent", which includes intent to buy or sell. Mahmud et al. (2016) also used manually-crafted features to infer potential purchase or recommendation intentions from Twitter.

Recently, convolutional and recurrent neural networks (CNN, RNN) have proven to be effective for different text processing tasks, e.g., (Kalchbrenner et al., 2014; Kim, 2014; Bahdanau et al., 2015; Cho et al., 2014; Hermann et al., 2015). They learn features automatically. Ding et al. (2015) applied a CNN to identify consumption intention from a single tweet. Korpusik et al. (2016) employed a simple average of word embeddings to model tweets and used a long short-term memory network for purchase prediction based on a user's tweet sequence. Both Ding et al. and Korpusik et al. focused on a binary classification task, rather than finer-grained multi-class AIDA purchase stages our models identify. And both works used a relatively balanced dataset, thus avoiding the difficult but more realistic classification task on strongly imbalanced data.

3 Task and Data

3.1 Purchase Stage Classification

Following the AIDA model (Lewis, 1903; Dukesmith, 1904; Russell, 1921), we regard the following purchase stages: Awareness (A), Interest (I), Desire (D) and Action ('bought' action in our case, thus we use the abbreviation B). In addition, we include a class with a negative sentiment: Unhappiness (U). We use this class for any expression of unhappiness with a product, before or after buying it. Table 1 provides examples for the different purchase stages. Although it is possible that a user may express unhappiness and an AIDA stage simultaneously, this occurred in only 15 tweets out of over 100k total. The task we focus on in this paper is purchase stage classification, i.e. distinguishing the different purchase stages for individual tweets in a given tweet sequence.

3.2 Dataset Creation

Data Collection. For a dataset, we focus on public Twitter tweets. Twitter data for purchase prediction was also collected by Korpusik et al. (2016). They used hand-crafted regular expressions to identify tweets indicating that a user may have bought or wanted a product. However, their dataset was biased towards bought/want tweets and their patterns covered only a subset of possible bought/want phrases.

To create a more "real-world" set, we scraped web sites for mobile phones, tablets and watches available in 2016, collecting 98 model names. The full product names and relatively distinct model

names (e.g, 'iPad' but not 'one' as in HTC One) formed queries to the Twitter search API. The tweets were filtered for spam using the URL features from (Benevenuto et al., 2010) and spam words. User timelines for the remaining users were collected and the users filtered for spammers using all their tweets.

Annotation. Tweets containing at least one product mention were labeled with the AIDB+U purchase stages defined above, and those which do not express one of these stages were annotated with an artificial class 'N'. Two annotators were given examples of each of the AIDB+UN categories. They first individually labeled the tweets. Cohen's kappa between the annotators was 0.30. For tweets that both annotators labeled with any of AIDBU, Cohen's kappa was 0.77. In a second pass, the annotators discussed the tweets where they disagreed and agreed on a final label.

Tweet Sequences. We regard all tweets from one user as one sequence (temporally ordered). However, if the temporal distance between two successive tweets is more than two months, we split them into two sequences. This maximum distance has been chosen heuristically after a manual analysis of tweets and their time stamps.

Statistics. In total, we annotated 106,474 tweets from 3,000 users. After splitting the tweet sequences (see above), we obtained 10,277 sequences. The class distribution is as follows: A: 0.23%, I: 0.65%, D: 1.11%, B: 0.90%, U: 0.50%, N: 96.61% In our experiments, we only classify IDB+UN because class 'A' has very few samples.

4 Model

We propose to use a hierarchical NN (see Figure 1) for purchase stage identification. In our experiments, we compare its components at the different hierachy levels with alternative choices. Unlike most previous work on purchase prediction, we do not use hand-crafted features to avoid expensive data preprocessing and manual feature design.

First, we represent each word by its embedding, skipping unknown words. The embeddings have been trained with word2vec (Mikolov et al., 2013) on Twitter data (Godin et al., 2015).[1]

Next, we compute a tweet representation that models word order. We apply convolutional filters which are slid over the sentence. Afterwards, 3-max pooling (Kalchbrenner et al., 2014) extracts the most relevant scores.

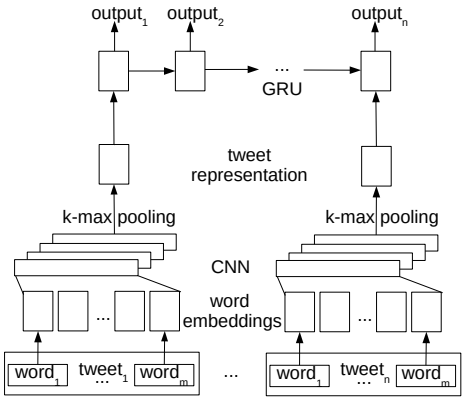

Figure 1: RCRNN: hierarchical neural network for purchase stage identification

Finally, we feed the representations of tweets by a user into a sequence model, i.e. a unidirectional NN with gated recurrent units (GRU) (Cho et al., 2014).[2] Thus, the model can learn patterns across tweets, such as "a user might first express interest in a product before buying it but not vice versa".

4.1 Dealing with Imbalanced Data

The dataset statistics show that the data is highly imbalanced. Users talking about products are not necessarily interested in buying them. Instead, they might write about their experience or mention that someone else has bought a product. To cope with the imbalanced labels, we propose to use a ranking layer. In our experiments, this approach outperforms traditionally used class weights.

Class Weights. If the ground truth is a non-artificial class, the error of the model is multiplied by $w > 1$. With gradient descent, the parameter updates after a false negative prediction are larger, penalizing the model more. The weight w_i for class i is proportional to the inverse class frequency f_i: $w_i \propto \frac{1}{f_i}$. The weights are normalized so that the weight for class 'N' is 1.

Ranking Loss. dos Santos et al. (2015) introduced the following ranking loss function:

$$\begin{aligned} L = &\log\left(1 + \exp(\gamma(m^+ - s_\theta(x)_{y^+}))\right) \\ &+ \log\left(1 + \exp(\gamma(m^- + s_\theta(x)_{c^-}))\right) \end{aligned} \quad (1)$$

[1] With the public Google News embeddings, we got consistently worse results, probably because of the domain mismatch and the higher number of out-of-vocabulary words.

[2] We have also experimented with bidirectional GRUs but observed that they performed worse. We assume that this might change with more training data.

$s_\theta(x)_{y^+}$ is the score for the correct label y^+ and $s_\theta(x)_{c^-}$ is the score for the best competitive class c^-. m^+ and m^- are margins. The function aims to give scores greater than m^+ for the correct class and scores smaller than m^- for the incorrect classes. The factor γ penalizes errors.[3] The function is especially suited for artificial classes (like our 'N' class) for which it might not be possible to learn a specific pattern: If $y^+ = N$, only the second summand is evaluated. During test, 'N' is only chosen if the scores for all other classes are negative. This lets the model focus on the non-artificial classes and is the reason why we investigate this loss function in the context of data which is imbalanced between AIDB+U and 'N'.

5 Experiments and Results

Due to the high class imbalance in our dataset, we use the macro F1 of the non-artificial classes as our evaluation measure. We implement the NNs with Theano (Theano Development Team, 2016) and the non-neural classifiers with scikit-learn (Pedregosa et al., 2011).

For training the NNs, we use stochastic gradient descent and shuffle the training data at the beginning of each epoch. We apply AdaDelta as the learning rate schedule (Zeiler, 2012). The hyper-parameters (number of hidden units, number of convolutional filters, and convolutional filter widths) are optimized on dev. We apply L2 regularization with $\lambda = 0.00001$ and early-stopping on the dev set. To avoid exploding gradients, we clip the gradients at a threshold of $t = 1$.

5.1 Data Preprocessing

To preprocess the tweets, we apply the publicly available scripts from Xu et al. (2016)[4] which use twokenize (Owoputi et al., 2013) for tokenization and perform some basic cleaning steps, such as replacing URLs with a special token or normalizing elongated words. Then, we split the data by user into training, development (*dev*) and test sets (80,10,10%). To reduce the class imbalance, we randomly subsample 'N' tweets in the training set. Table 2 provides statistics for the final dataset.

5.2 Experiments

Baseline Models. In addition to a random guessing baseline, we use two non-neural baseline models: A logistic regression classifier *(LR)* and a linear support vector machine *(SVM)*. For both models, the tweets are represented by 1-gram, 2-gram and 3-gram bag-of-word *(BOW)* vectors. Table 3 shows that the RCRNN clearly outperforms non-neural models.

[3] We set m^+ to 2.5 and m^- to 0.5 as in (dos Santos et al., 2015) but tune γ on dev.

[4] https://github.com/stevenxxiu/senti/tree/master/senti

		train	dev	test
	# tweets	16,715	2,371	2,312
	# tweet sequences	3,938	559	546
label distr.	# class I	496	74	89
	# class D	864	173	145
	# class B	721	129	112
	# class U	393	80	61
	# class N	14,241	1,915	1,905

Table 2: Dataset statistics after preprocessing

Model	dev F1	test F1
Random Guessing	4.17	4.02
BOW SVM	43.03	43.97
BOW LR	40.25	42.32
RCRNN	**51.65**	**51.39**

Table 3: RCRNN vs. baseline models

Impact of RCRNN Components. We first investigate CNN against two other methods for calculating tweet representations (Table 4): (1) Averaging word embeddings *(Average)* (Korpusik et al., 2016; Le and Mikolov, 2014) and (2) a bidirectional GRU with attention *(GRU+att)*. For the GRU, we use the equations provided in (Cho et al., 2014). For each intermediate hidden layer x_i of the GRU, we calculate the attention weight α_i with a softmax layer:

$$\alpha_i = \frac{exp(V^T x_i)}{\sum_j exp(V^T x_j)} \quad (2)$$

where V is a parameter of the model that is initialized randomly and learned during training. We then use the weighted sum of all hidden layers as the tweet representation.

GRU+att and CNN clearly outperform Average which can neither take word order into account nor focus on relevant words. Also, CNN outperforms GRU+att.

Next, we show the positive impact of GRU as a tweet sequence model by replacing it with models that do not use sequential information. In particular, we use a simple feed-forward *(FF)* model

Tweet representation model	dev F1	test F1
Average	44.01	45.21
GRU+att	49.52	50.75
CNN (RCRNN)	**51.65**	**51.39**

Table 4: Impact of tweet representation model

Tweet sequence model	dev F1	test F1
FF, no hidden layer	49.64	45.15
FF + hidden layer	51.11	48.73
GRU (RCRNN)	**51.65**	**51.39**

Table 5: Impact of tweet sequence model

Loss function	dev F1	test F1
CE	48.71	48.43
CE+weights	49.88	49.01
Ranking (RCRNN)	**51.65**	**51.39**

Table 6: Impact of ranking layer on RCRNN

(with and without a hidden layer) to predict the output label given only the current tweet representation calculated by a CNN. The results provided in Table 5 show that GRU outperforms the FF models. Thus, there is cross-tweet information which can be exploited for purchase stage prediction.

Finally, we investigate ways of dealing with imbalanced data: We replace the ranking layer of RCRNN with a cross-entropy *(CE)* loss with and without class weights (see Section 4.1). Table 6 shows that class weights improve CE but ranking performs best.[5] Adding class weights to the baseline SVM improves the model to 46.27 on dev and 50.89 on test. The performance on dev and test are both still worse than RCRNN. Thus, our experiments do not confirm previous studies which found that SVMs were superior to NNs on imbalanced data (Chawla et al., 2004).

To sum up, we observed that convolution provided the best tweet representation while a GRU was helpful to model tweet sequences. Ranking could best deal with class imbalance.

5.3 Analysis

Figure 2 shows the confusion matrix for RCRNN. Apart from confusions with 'N' which most probably result from the class imbalance, the model confuses neighboring labels, such as 'I' and 'D'. In total, over 90% of the confusions involve 'N'. This shows that the model is reasonably good at distinguishing the purchase stages and that the main difficulty is class imbalance. In future work, we will extend the investigation of this topic.

6 Conclusion

We defined a purchase stage identification task based on the AIDA model. We compared several

[5]This result is also consistent with Average and GRU+att as tweet representation models

ref \ hypo	N	I	D	B	U
N	1853	16	19	19	27
I	52	31	6	0	0
D	61	8	75	1	0
B	44	2	5	60	1
U	37	0	2	0	22

Figure 2: Confusion matrix on test set

neural and non-neural models of tweets and tweet sequences and observed the best performance using RCRNN, our ranking-based hierarchical network which uses convolution to represent tweets and gated recurrent units to model tweet sequences. Our results indicate that tweets indeed contain signals indicative of purchase stages which can be captured by deep learning models. Ranking was the most effective way to deal with class imbalance.

References

Sitaram Asur and Bernardo A. Huberman. 2010. Predicting the future with social media. In *2010 IEEE/WIC/ACM International Conference on Web Intelligence, WI 2010, Main Conference Proceedings*, pages 492–499, Toronto, Canada, August.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations (ICLR)*, San Diego, California, USA, May.

Fabricio Benevenuto, Gabriel Magno, Tiago Rodrigues, and Virgilio Almeida. 2010. Detecting spammers on Twitter. In *CEAS 2010 - Seventh annual Collaboration, Electronic messaging, Anti-Abuse and Spam Conference*, Redmond, Washington, July.

Johan Bollen and Huina Mao. 2011. Twitter mood as a stock market predictor. *Computer*, 44(10):91–94, October.

Nitesh V Chawla, Nathalie Japkowicz, and Aleksander Kotcz. 2004. Editorial: special issue on learning from imbalanced data sets. 6(1):1–6, June.

Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar, October. Association for Computational Linguistics.

Xiao Ding, Ting Liu, Junwen Duan, and Jian-Yun Nie. 2015. Mining user consumption intention from social media using domain adaptive convolutional neural network. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, pages 2389–2395, Austin, Texas, January.

Cicero dos Santos, Bing Xiang, and Bowen Zhou. 2015. Classifying relations by ranking with convolutional neural networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 626–634, Beijing, China, July. Association for Computational Linguistics.

Frank Hutchinson Dukesmith. 1904. Three natural fields of salesmanship. *Salesmanship*, 2(1):14, January.

Fréderic Godin, Baptist Vandersmissen, Wesley De Neve, and Rik Van de Walle. 2015. Multimedia lab @ acl wnut ner shared task: Named entity recognition for twitter microposts using distributed word representations. In *Proceedings of the Workshop on Noisy User-generated Text*, pages 146–153, Beijing, China, July. Association for Computational Linguistics.

Vineet Gupta, Devesh Varshney, Harsh Jhamtani, Deepam Kedia, and Shweta Karwa. 2014. Identifying purchase intent from social posts. In *Proceedings of the Eighth International Conference on Weblogs and Social Media, ICWSM 2014*, Ann Arbor, Michigan, June.

Karl Moritz Hermann, Tomás Kociský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015*, pages 1693–1701, Montreal, Quebec, Canada, December.

Bernd Hollerit, Mark Kröll, and Markus Strohmaier. 2013. Towards linking buyers and sellers: detecting commercial intent on twitter. In *22nd International World Wide Web Conference, WWW '13, Companion Volume*, pages 629–632, Rio de Janeiro, Brazil, May.

Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. 2014. A convolutional neural network for modelling sentences. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 655–665, Baltimore, Maryland, June. Association for Computational Linguistics.

Milad Kharratzadeh and Mark Coates. 2012. Weblog analysis for predicting correlations in stock price evolutions. In *Proceedings of the Sixth International Conference on Weblogs and Social Media*, Dublin, Ireland, June.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, Doha, Qatar, October. Association for Computational Linguistics.

Mandy Korpusik, Shigeyuki Sakaki, Francine Chen, and Yan-Ying Chen. 2016. Recurrent neural networks for customer purchase prediction on twitter. In *Proceedings of the 3rd Workshop on New Trends in Content-Based Recommender Systems co-located with ACM Conference on Recommender Systems (RecSys 2016)*, pages 47–50, Boston, MA, USA, September.

Niels Buus Lassen, Rene Madsen, and Ravi Vatrapu. 2014. Predicting iphone sales from iphone tweets. In *18th IEEE International Enterprise Distributed Object Computing Conference, EDOC 2014*, pages 81–90, Ulm, Germany, September.

Quoc V. Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014*, pages 1188–1196, Beijing, China, June.

Elias St. Elmo Lewis. 1903. Catch-line and argument. *The Book-Keeper*, 15:124–128, February.

Caroline Lo, Dan Frankowski, and Jure Leskovec. 2016. Understanding behaviors that lead to purchasing: A case study of pinterest. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 531–540, San Francisco, CA, USA, August.

Jalal Mahmud, Geli Fei, Anbang Xu, Aditya Pal, and Michelle X. Zhou. 2016. Predicting attitude and actions of twitter users. In *Proceedings of the 21st International Conference on Intelligent User Interfaces, IUI 2016*, pages 2–6, Sonoma, CA, USA, March.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. In *Proceedings of Workshop at 1st International Conference on Learning Representations (ICLR)*, Scottsdale, Arizona, USA, May.

Meredith Ringel Morris, Jaime Teevan, and Katrina Panovich. 2010. What do people ask their social networks, and why?: a survey study of status message q&a behavior. In *Proceedings of the 28th International Conference on Human Factors in Computing Systems, CHI 2010*, pages 1739–1748, Atlanta, Georgia, USA, April.

Olutobi Owoputi, Brendan O'Connor, Chris Dyer, Kevin Gimpel, Nathan Schneider, and Noah A. Smith. 2013. Improved part-of-speech tagging for online conversational text with word clusters. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*,

pages 380–390, Atlanta, Georgia, June. Association for Computational Linguistics.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake VanderPlas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Edouard Duchesnay. 2011. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12:2825–2830.

J Ramanand, Krishna Bhavsar, and Niranjan Pedanekar. 2010. Wishful thinking - finding suggestions and 'buy' wishes from product reviews. In *Proceedings of the NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text*, pages 54–61, Los Angeles, CA, June. Association for Computational Linguistics.

C.P. Russell. 1921. How to write a sales-making letter. *Printers' Ink*, 115:49–56, June.

Theano Development Team. 2016. Theano: A Python framework for fast computation of mathematical expressions. In *arXiv:1605.02688*.

Armando Vieira. 2015. Predicting online user behaviour using deep learning algorithms. In *arXiv:1511.06247*.

Steven Xu, HuiZhi Liang, and Timothy Baldwin. 2016. Unimelb at semeval-2016 tasks 4a and 4b: An ensemble of neural networks and a word2vec based model for sentiment classification. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 183–189, San Diego, California, June. Association for Computational Linguistics.

Matthew D. Zeiler. 2012. ADADELTA: An adaptive learning rate method. In *arXiv:1212.5701*.