Negative Purchase Intent Identification in Twitter

Samed Atouati
Telecom Paris, IP Paris
France
samed.atouati@telecom-paris.fr

Xiao Lu
BNP Paribas Asset Management
Paris, France
xiao.lu@bnpparibas.com

Mauro Sozio
Telecom Paris, IP Paris
France
mauro.sozio@telecom-paris.fr

ABSTRACT

Social network users often express their discontent with a product or a service from a company on social media. Such a reaction is more pronounced in the aftermath of a corporate scandal such as a corruption scandal or food poisoning in a chain restaurant. In our work, we focus on identifying *negative purchase intent in a tweet*, i.e. the intent of a user of not purchasing any product or consuming any service from a company. We develop a binary classifier for such a task, which consists of a generalization of logistic regression leveraging the locality of purchase intent in posts from Twitter. We conduct an extensive experimental evaluation against state-of-the-art approaches on a large collection of tweets, showing the effectiveness of our approach in terms of F1 score. We also provide some preliminary results on which kinds of corporate scandals might affect the purchase intent of customers the most.

KEYWORDS

social media, neural networks, purchase intent, classification, company scandal, hashtag segmentation

ACM Reference Format:
Samed Atouati, Xiao Lu, and Mauro Sozio. 2020. Negative Purchase Intent Identification in Twitter. In *Proceedings of The Web Conference 2020 (WWW '20), April 20–24, 2020, Taipei, Taiwan.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3366423.3380040

1 INTRODUCTION

Social network users often express their discontent with a service or a product from a company on Twitter. This happens for example when a flight gets canceled, a baggage gets lost, or there is a long queue at the post office. In the aftermath of a company scandal, such as a corruption scandal or food poisoning in a chain restaurant, such a reaction is more pronounced with messages of the kind "never again with this company" being relatively frequent.

In our work, we aim at identifying tweets expressing the intention of a user of not purchasing any product or consuming any service from a given company. We shall refer to such an intention as *"negative purchase intent"* (negative PI). In particular, our goal is to develop a binary classifier answering the question: does the text contain a negative PI?

WWW '20, April 20–24, 2020, Taipei, Taiwan

ACM ISBN 978-1-4503-7023-3/20/04.
https://doi.org/10.1145/3366423.3380040

Our main motivation for developing a negative PI classifier is to study how events affect the purchase intent of potential customers: would a data breach in the customer database have larger or smaller impact than an issue in the customer service of a company?

Previous work focuses on the task of identifying the intent of purchasing a product or consuming a service ([20],[11],[6],[3]). However, PI classifiers cannot be easily adapted to solve our problem. In particular, posts that are not classified as PI do not necessarily express a negative sentiment, while a negative sentiment does not necessarily translate into a negative PI. Moreover, PI classifiers ignore the presence of sentences such as "never again" which likely signal a negative PI.

There have been significant efforts in the area of sentiment analysis in recent years ([18]) however, a negative sentiment does not necessarily indicate a negative PI and viceversa. For illustrative purposes, consider the examples provided in Table 1, in particular the second and third examples. The post "I enjoyed my last flight with @British_Airways. But I can't afford their price anymore." expresses a positive sentiment, however, the users is unlikely to fly again with British Airways. On the other hand, the post "I always fly BA for Europe travels. Sometimes the staff are rather rude." contains a negative sentiment, however, it is unclear whether the user will fly again with British Airways or not.

One of the main challenges to tackle when developing a classifier for our problem is that negative PI is expressed using a relatively rich vocabulary, which poses non-trivial challenges in feature selection. This problem is exacerbated by the lack of labeled data. Moreover, in the case of Twitter, negative PI might be expressed solely by means of hashtags, while tweets notoriously contain typos or noise.

Our approach involves a fine-grained representation of the text associated with a model that exploits the locality of purchase intent. In particular, a negative PI is often localized in 1-2 words as well as a few other words in their immediate proximity (neighborhood) such as "never fly again with" or "let's boycott". We leverage the locality of a negative PI by a variant of logistic regression which is generalized so as to take into account the neighborhood representation of a tweet. We then conduct an experimental evaluation on a large collection of tweets showing that our approach outperforms state-of-the-art approaches in terms of F1 score. Finally, we perform a case study providing some preliminary insights on which kind of corporate scandals might have more significant impact on the purchase intent of customers.

Problem Definition and Motivations. We define negative purchase intent (negative PI) as the explicit expression of a user of not consuming or purchase a given item or a service from a company. Negative PI can be expressed using a relatively rich vocabulary,

Table 1: Negative sentiment and negative purchase intent.

Neg. sent. and Neg. PI	The staff members were awful. I will not be booking with them again.
Pos. sent. and Neg. PI	I enjoyed my last flight with @British_Airways. But I can't afford their price anymore.
Neg. sent. but not Neg. PI	I always fly BA for Europe travels. Sometimes the staff are rather rude.
Pos. sent. but not Neg. PI	The flight was very enjoyable. Will make sure to fly with you again.

such as "let's boycott...", "I will avoid...", "I will never buy again...", "...never flown since". Observe that a sentence might express a negative sentiment towards a company but not a negative PI and vice versa, or even a positive sentiment and a negative PI. Table 1 provides an example for each possible combination.

Our goal is to develop a binary classifier answering the question: does the text contain a negative PI? This is a natural language understanding task that has not been addressed so far, to the best of our knowledge. Although we focus on Twitter, our approach can be adapted to other social media or other data sources as well. Our main motivation is to study how events affect the purchase intent of potential customers, with the long-term goal of characterizing what kinds of events affect customer purchase intent the most. For example, would a data breach in the customer database have larger or smaller impact than an abuse toward a customer? Our work aims at shedding some light on customer behavior which finds applications in brand reputation management, financial investments, and many others. For example, a company might handle a scandal depending on whether such a scandal is supposed to significantly affect the customer purchase intent or not. Furthermore, a scandal which is likely to affect significantly customer purchase intent might provide some investment opportunities.

The rest of the paper is organized as follows. In Section 2, we summarize the related work. In Section 3, we present our approach which is evaluated against a few baselines and the state-of-the-art in Section 4. We recap and summarize our conclusions in Section 5.

2 RELATED WORK

Due to the wealth of information present in social media (social networks, forums, micro-blogging), identification of PI, either explicit or implicit, has attracted the attention of the research community. In [7], Hamroun et al. use lexico-semantic patterns to extract PI patterns, in a parallel computing framework. In [20], Wang et al. use a bootstrapping method to extract intent indicators by starting with a small seed set. They use the final set of key indicators to formulate categorical PI identification as a graph-based semi-supervised learning problem. In [6], the authors exploited various lexical and grammatical features, as well as sentiment analysis, to build an SVM classifier for PI identification trained on Quora posts and tested on Yahoo ones.

Company scandals have also attracted the interest of researchers from diverse fields (data science, sociology, management), as learning about the mechanics of public reaction helps the companies mitigate the effect of scandals on their image, and by extension, their business. In [4], the author studies the Jetblue Airways Valentine day's crisis of 2007, and showed that the positive effect of the CEO's apology on YouTube on the public's perception of the company. In [14], the authors studied sentiment evolution regarding Domino's Pizza in the aftermath of their 2009 scandal. They observed that although the positive sentiment didn't increase after the CEO's apology on YouTube, the negative sentiment decreased. This indicates that taking responsibility during a company crisis lessens reputation damage, which confirms [4]'s findings. The Volkswagen scandal of 2015 also attracted attention from the research community. In [1], the authors study the public opinion's evolution regarding VW and provide interesting insights regarding the financial and industrial impacts of the crisis on VW during the 2012-2016 period. In [16], the authors tackle the same scandal and postulate that the extreme negativity may be due to VW not communicating on the crisis.

Transfer learning has also been studied in the context of PI. In [3], Ding et al. use a CNN architecture to learn PI identification for one product category, and retrain the last layer of their model to learn PI identification for another product category. And in [23], the authors use an EM-like scheme to train a Naive Bayes classifier to identify PI for 4 tech products dnomains. In [15], the authors use a set of PI queries to harvest tweets and use the Word2Vec model of [12], to build a vector variable for each day. This variable was used to predict the future Consumer Spending Index, alongside its lagged values.

The problem studied in this work is negative PI identification : the intent not to buy/consume from a given company, and how scandals can affect it. To the best of our knowledge, negative PI has not been studied yet, with most of the work focusing on purchase intent identification. Along this line of work, the most relevant ones to our approach are the approach in [11] and the approach in [6]. In [11], the authors used a (Subject, Verb Phrase, Object) pattern flagging scheme to identify PI. In [6], the authors used an exhaustive set of features, including a sentiment score and syntactical pattern rules, to represent the social posts. It does not seem to be trivial to adapt the work for PI identification to our problem. In particular posts that are not classified as PI do not necessarily contain a negative PI.

3 OUR APPROACH

We organize the description of our approach into the three following sections: data collection and labeling, data representation, and cost function. For reproducibility and self containment, we strive to provide all relevant information. We call our approach NeighLoR, which stands for neighborhood-based logistic regression, in that, it can be seen as a variant of logistic regression taking into account the neighborhood representation of tweets. The notebooks for reproducibility of experiments can be found at https://github.com/NPIDT/NegPI.

3.1 Data Collection and Labeling

We focus our attention on two main events which had significant impact on the user activity in social media and the public opinion in general. The two events involve the British Airways company: the first one is about the hack of approximately 380K credit cards from

customers of the airline company, which took place in September 2018, while the second one is about its flight attendants deplaning an Indian family because their 3 years old child was crying, which took place in 09-08-2018. We collected approximately 170K tweets from August 1st, 2018 to October 31rd 2018 using the query 'british airways' on the Twitter API. This dataset contains therefore tweets referring to both these two events.

One of the major challenges we faced was to obtain a sufficiently large number of labeled tweets which could be used for training and for validation. An additional challenge was due to the fact that only a tiny fraction of tweets express a negative purchase intent, resulting in a class imbalance problem. We tackled those challenges by producing two more datasets containing a subset of tweets which could be labeled manually and where the class imbalance problem is mitigated.

In particular, we obtain 11684 tweets from the British-Airways dataset as follows. We first extract all those tweets containing words such as "never" and "not" obtaining 3.1K tweets in total. This maximizes the chances of retaining tweets with negative purchase intent. The remaining tweets have been sampled uniformly at random from the remaining tweets. With the help of five volunteers, we manually labeled the 11684 tweets as either negative PI or not, obtaining 1096 negative PI tweets, and 10588 non-negative PI tweets.

Manual evaluation. With the help of five volunteers we labeled manually 11684 tweets from the British-Airways dataset. The volunteers are PhD students in the field of machine learning or information extraction, who could provide a reliable labeling. Each volunteer was provided with the definition given in Section 1, while he/she was given a chunk of tweets to label as containing a negative PI or not. We made sure to differentiate clearly between negative PI and negative sentiment, while providing a few examples where the negative PI was expressed in a hashtag.

We then use the labeled tweets for training and validation, in a cross-validation setting. In order to provide a rigorous evaluation and in particular to make sure that our method performs well on an arbitrary dataset, we also evaluate our method on the original collection of tweets (containing more than 300k tweets) by a manual evaluation on a sample chosen uniformly at random. This allows us to evaluate our approach with a 0.95-confidence interval.

Table 2 and 3 summarize the statistics for our dataset.

Table 2: Summary statistics for collected data.

Company	Period	Total	Labeled
British Airways	08-2018 to 10-2018	170K	11684

Table 3: Events in the dataset.

Company	Event	Date
British Airways	Deplaning of Indian family	09-08-2018
British Airways	Data breach	07-09-2018

3.2 Data Representation

The data representation can be decomposed into the following steps: 1) Preprocessing, 2) Part-Of-Speech tagging, 3) Hashtag segmentation, 4) Feature selection, 5) Neighborhood construction.

3.2.1 Preprocessing. We perform the following standard preprocessing techniques: We remove URLs; we collapse repeated characters in words by leaving a maximum of two successive repetitions of the same character; we remove numbers and words that contain them; we expand shortened expressions to their full form such as "haven't" → "have not", "I've" → "I have"; we perform lemmatization; we replace the various designations of the company of interest with a unique token. We also replace the various designations of other companies with a unique token.

3.2.2 Part-Of-Speech tagging. We use the CMU POS tagger [13] to perform our POS tagging, which is one of the state-of-the-art POS taggers specifically designed for tweets.

Table 4: Hashtag Segmentation: Examples

Initial hashtag	Segmented hashtag
#notsokeenonflyingbaagain	not so keen on flying ba again
#neveragainonba	never again on ba
#dontflyba	dont fly ba
#myluggageisahavingabetter vacationthanme	myluggageisahavingabetter vacationthanme

3.2.3 Hashtag segmentation. Hashtags convey important information when identifying purchase intent. According to our experiments, negative PI is communicated solely by means of hashtags in 16.5% of tweets (in our british airways dataset). In other words, if hashtags were neglected in those tweets, we would not be able to classify those tweets as expressing negative PI. In order to fully leverage the information carried by the hashtags, each hashtag has to be decomposed into its list of meaningful words. This procedure is called hashtag segmentation. Table 4 shows an example for the british airways dataset. We implement the approach developed in [19] to segment hashtags, starting from the POS tagged tweets. Note, however,that this method is not bullet-proof and in particular it suffers from the presence of noise and typos in tweets which is Table 4's last example.

3.2.4 Feature selection. In this step, we select the most relevant words for our classification task. First, we run the CMU POS tagger [13] on the tweets, and retain only adverbs, verbs and adjectives with at least 5 occurrences. We then perform lemmatization on them using *NTLK*'s WordNet lemmatizer [17]. After that, we select the words which are highly correlated with the negative PI class. To this end, we compute the Pointwise Mutual Information (PMI) between word occurrences and the negative PI class. Words with large PMI have high correlation. Formally, given a tweet t, we let $y_t = 1$ if t expresses negative purchase intent, while we let $y_t = 0$ otherwise. We denote with $w \in t$ the fact that a word w appears in the tweet t. For every tweet t, for every word w, we let

$$PMI(y_t = 1, w) := \log\left(\frac{P(y_t = 1, w \in t)}{P(y_t = 1) \cdot P(w \in t)}\right). \quad (1)$$

Since $P(y_t = 1)$ is constant and the logarithm is strictly increasing, it suffices to rank the words according to their $P(y_t = 1|w \in t)$ value. To prevent that rare words be penalized in the ranking, we use the lower bound of the symmetric Gaussian confidence interval at the 95% confidence level:

$$s(w) := \widehat{P}(y = 1|w) + z_\alpha \times \sqrt{\frac{\widehat{P}(y = 1|w) \cdot \widehat{P}(y = 0|w)}{N_w}} \quad (2)$$

with $N_w = \{t; w \in t\}$, $\alpha = 0.025$, and z_α the 2.5% quantile of the normal distribution. In this case, $z_\alpha = -1.96$. $\widehat{P}(y_t = 1|w \in t)$ is the estimator of the conditional probability of the negative PI class computed over the set of tweets T, that is

$$\widehat{P}(y_t = 1|w \in t) := \frac{|\{t; y_t = 1, w \in t\}|}{|\{t; w \in t\}|}. \quad (3)$$

The underlying assumptions are that the tweets constitute independent observations, while the variable $X_w(t) := 1_{\{y_t=1|w\in t\}}$ follows a Bernoulli distribution.

Furthermore, we use the dependency parsing tool from the Python package *Spacy* [8] to extract negation relationships between words. Such a tool allows us to extract features of the kind "not buy" from "'never ever buying"' or "never fly" from "never will I fly", which are represented by one-hot-encoding. We also add the feature "No-BA" which replaces any mention of an airline company other than British Airways. This shall be useful in discriminating whether a negative PI is directed towards another airline company. We select the top 20% words as well as the bottom 20% words in the ranking, which help us discriminating between the negative PI class and the rest.

3.2.5 Neighborhood construction. In our experiments, we observe that the expression of a negative PI is usually localized in a relatively short part of tweets. For example in the tweet "We were left in an empty bus at Funchal airport then when they realised and came to get us to board the plane lots of passengers glared and tutted, then arrived at Bristol to broken stair and missing wheelchair.....**never flown since**", the negative PI is expressed in a short sentence (highlighted in bold). This motivates us to split a tweet into several (overlapping) parts (called *neighborhoods*), with each of them potentially containing a negative PI. This allows us to take into account the context of each word, while filtering out noise or non-relevant information.

We define the l-neighborhood of a word w_k in a tweet as the window of $2l+1$ words whose center is $w_k : [w_{k-l}, \cdots, w_k, \cdots, w_{k+l}]$. For each verb and adverb as a center, a neighborhood is built, which is truncated to the available size if there are not enough words. Observe that neighborhoods might contain adjectives or bigrams (obtained in the feature-selection step), or segmented hashtags. Given a tweet t and d features, we obtain a matrix representation $X_t \in R^{N_t \times d}$ with

$$X_t^{i,j} = \begin{cases} 1, & \text{if feature token } w_j \text{ in neighborhood } i \text{ of tweet } t \\ 0, & \text{otherwise,} \end{cases} \quad (4)$$

where N_t denotes the number of neighborhoods in t.

3.3 Cost function

When applying machine learning to classification tasks, the cost function can be decomposed into two elements: the probability hypothesis function and the classification error function. In the case of logistic regression, the hypothesis function is

$$h_\beta(X) := \sigma(\beta^T X) := \frac{1}{1 + \exp(-\beta^T X)}\ \beta, X \in R^d,$$

while the classification error function is the cross-entropy between target y and estimated probability of negative PI class $\widehat{p}$:

$$CE(y, \widehat{p}) = -y \cdot \log(\widehat{p}) - (1 - y) \cdot \log(1 - \widehat{p}). \quad (5)$$

This leads to the following cost function:

$$C(X, y; \beta) = -y \cdot \log(\sigma(\beta^T X)) - (1 - y) \cdot \log(1 - \sigma(\beta^T X))), \quad (6)$$

For our task, we modify the hypothesis function as follows:

$$h_\beta(X_t) = \max_{1 \le i \le N_t} \sigma(\beta^T X_t^i), \quad (7)$$

where N_t denotes the number of neighborhoods in tweet t. In other words, we compute the sigmoid function for every neighborhood in a tweet, while computing the maximum value among such neighbors. This captures the intuition that a negative PI is often localized in a neighborhood. We derive the final cost function as follows:

$$L(X, y; \beta) = \frac{1}{|T|} \sum_{t \in T} CE(y_t, \max_{1 \le i \le N_t} \sigma(\beta^T X_t^i)) \quad (8)$$

with CE being the cross-entropy loss as defined in Equation (5). One can easily show that $L(X, y; \beta)$ is locally convex, however, it is neither convex nor quasi-convex. As a result, first order methods may not find a global minimum, while we can still find local minima by means of gradient descent.

The cost function proposed can also be interpreted as a neural network over the neighborhoods representation with: a) A linear layer with a sigmoid activation function over inputs of unknown first dimension $X_t \in R^{N_t \times d}$, with d fixed and N_t unknown; b) a max-pooling layer(see for example [21]); c) a cross-entropy loss. Compared to other deep learning approaches, our approach, while being less complex, has the advantage of being relatively easy to interpret in contrast with state-of-the-art deep learning methods.

Our approach can also be interpreted as a special-case CNN [10] unit, where the representation fed to the max-pooling part remains a simple bag-of-words, and where the stride of the CNN's max-pooling is defined by word function rather than by a fixed step.

4 EXPERIMENTAL RESULTS

4.1 Settings

We compare our approach against the approaches developed in [11] and [6], as well as against the following natural baselines: 1) an LSTM-based approach [5]; 2) Google BERT model (BERT) [2]; two approaches based on Word2Vec [12], namely, 3) the pre-trained model from Google (W2Vec-PT); and 4) a model we train on our dataset using the *Gensim* package (W2Vec-Cust). In order to ensure a fair comparison, all approaches are allowed to be trained for at most two hours.

For our approach NeighLoR, we use the following settings and parameters. We run gradient descent on the cost function $L(X, y; \beta)$

defined in Equation 8 with a maximum of 5000 iterations, constant learning rate of $\alpha = 10$, and a tolerance of 10^{-4} on the gradient norm. We initialize the weights in the model as follows: $\beta_0 = logit(\bar{y})$ as the initial intercept and $\forall i, \beta_i = 0$. Each neighborhood consists of a 7-word window.

Hamroun et al.'s approach [11] consisted in spotting patterns that are usually used in expressing purchase intent. Gupta et al. [6] on the other hand, selected n-gram features based on information gain and used the linear SVM model to learn a classifier on purchase intent. They used the *Alchemy* API for sentiment analysis. Since it is not available anymore, we used the VADER sentiment analysis tool [9] instead. We also add the split hashtags into the text fed to Gupta's approach. However, we do not use their WordNet similarity part as it is unwise in our case, where words that are vastly different from a dictionary standpoint are used to mean the same thing : 'fly' and 'use' mean the same thing for our task on British Airways for example.

Regarding the Word2Vec benchmark, we compute the average embedding of every tweet's words, restricted to meaningful tags (verbs, adverbs, adjectives, and nouns), as the features for a logistic regression, we train the resulting model in a 10-fold cross validation. We also combine the word embeddings with our approach, by having an average embedding of the words in a neighborhood as its representation. This is done for both the pre-trained Word2Vec and our custom version. The custom Word2Vec is trained for British Airways using 330K tweet.

LSTM consists of: an embedding layer of dimension $d = 50$; an LSTM layer with $c = 32$ cells; a linear layer with dimension $l = 256$ + ReLU activation; a dropout layer with a dropout probability of $p_d = 0.5$; a linear layer with a sigmoid activation function. We keep 1500 most frequent words for the embedding layer, which gave the best performance in terms of F1-score, while we train our model using the *Keras* API.

As for the BERT model, we consider both the github repository from Google [1] and the pytorch-transformers library [2]. We only report the results of the latter one, as it performs best in our experimental evaluation. We use the cased model with 12 transformer layers, 768 hidden size in the transformer, 12 the number of heads in the multi-attention layer, while we train a linear + softmax additional layer to be applied on the final sentence embedding. The training is done over 10 epochs with a dimension of 50. The number of epochs has been determined so that the overall training requires around two hours. We remark that in case BERT is allowed to train for up to 5 hours (corresponding roughly to 20 epochs), it achieves the remarkable F1-score of 81.8 outperforming our approach. We use the Word2Vec features to train a logistic regression classifiers. For the sake of fair comparison, we add the split hashtags to the word sequences that serve as input to the various models tested.

4.2 Evaluation methods

We evaluate all approaches in terms of F1-score and ROC. Measuring the accuracy of the approaches would not provide any meaningful information, given the high class imbalance in our dataset. In particular, a classifier that labeled all tweets as non-negative PI would have an accuracy of 90% or higher. Therefore, accuracy is not reported. We consider two different evaluation methods. First, we perform a 10-fold cross-validation over the labeled data, which consists of approximately $11k$ tweets related to British Airways, see Table 2.

Then, we evaluate our approach on the whole collection of tweets, which consists of approximately 330k tweets for British Airways. This last step is more delicate, in that, it would be very tedious and cumbersome to manually evaluate more than 330k tweets. We proceed as follows. We use the labeled data for training a model we use to label all 330k additional tweets. Then, we draw a sample uniformly at random from these tweets and compute manually the errors of our model in that sample. To this end, the 5 volunteers were given the same instructions specified in Section 3.1. This allows us to estimate the precision and recall of our classifier with a 0.95 confidence interval as follows.

The precision on the whole dataset can be estimated as a function of the precision of our classifier in the sample and then use standard statistic tools, such as Wilson score interval [22], so as to compute a 0.95-confidence interval. The recall R is defined as $R = \frac{TP}{TP+FN}$, with TP being the true positives, and FN being the false negatives. By evaluating our approach on the sample and using standard statistic tools, we obtain a 0.95-confidence lower bound for TP and a 0.95-confidence upper bound for $TP + FN$, from which we derive a lower bound on R on the whole dataset. We have $TP = n_1 \cdot \widehat{p_1}$ and $TP + FN = n_1 \cdot \widehat{p_1} + n_0 \cdot (1 - \widehat{p_0})$, where $\widehat{p_1}$ and $\widehat{p_0}$ are the estimates obtained on the sample for the precision on the positive class, and the precision on the negative class, respectively; and n_1/n_0 is the number of tweets labeled as positive/negative class respectively. We replace the estimate of $TP/TP + FN$ by their lower/upper bound in order to obtain the recall estimate.

4.3 Results on the British Airways dataset

4.3.1 Cross-validation. We perform a 10-fold cross-validation to evaluate all methods in terms of F1 score and ROC. For the error term of [6]'s support vector machine (C, we use for each fold a grid search for values range $(10^{-4+k/2})_{k=0,\cdots,16}$ and keep the model with largest training F1 score to test on the fold's test set. The results are shown in Table 5.

Table 5: 10-fold cross validation for British Airways.

Method	P	R	F1	ROC
W2Vec-PT + LogReg	71.28	47.9	57.2	89.24
W2Vec-Cust + LogReg	72.76	41.42	52.68	88.88
LSTM	69.58	59.45	60.38	91.86
BERT	**94.77**	60.83	73.81	95.03
[11]	62.19	65.77	63.93	NA
best model using [6]	87.02	59.74	70.85	92.43
NeighLoR	84.48	**77.56**	**80.79**	**97.08**

We can see that our model significantly outperforms most of the other methods in terms of all metrics, except for precision where the approach in [6] performs better. However, this comes at the cost of a significantly lower recall. In fact, our model significantly

[1] https://github.com/google-research/bert
[2] https://pytorch.org/hub/huggingface_pytorch-transformers/

outperforms all other approaches both in terms of F1-score and ROC AUC. On the other hand the somehow disappointing results of W2Vec-Cust, LSTM, and BERT might be due to the fact that these models unravel their full potentials when they are trained on a much larger collection of labeled data. Unfortunately, it does not seem to be trivial to obtain a large collection of labeled data for our classification task.

Next, we consider several variants of our approach. In particular, we consider two variants where our approach is combined with Word2Vec (pretrained and customized), as well as a vanilla logistic regression approach. Results are shown in Table 6. We can see that our approach does not benefit much from the Word2Vec embeddings. This might be due to the fact that the amount of labeled data might be relatively limited for Word2Vec to deliver good results. Moreover, Word2Vec might not work well when dealing with whole sentences. We also see that our approach performs significantly better than vanilla logistic regression.

Table 6: 10-fold cross-validation for the variants of our model.

Method	P	R	F1	ROC
W2Vec-PT + **NeighLoR**	70.14	76.4	71.04	95.32
W2Vec-Cust + **NeighLoR**	38.16	34.67	32.65	75.33
LogReg	76.98	64.23	69.93	93.11
NeighLoR	**84.48**	**77.56**	**80.79**	**97.08**

4.3.2 Estimating precision and recall on the whole dataset. As discussed in Section 4.2, we evaluate our approach on a sample of 300 tweets chosen uniformly at random from the negative PI class, as well as, 300 tweets chosen uniformly at random from the other class. The tweets in the sample were labeled manually with the help of 5 volunteers who were given the same instructions specified in Section 3.1. Table 7 shows the matrix so obtained.

Table 7: Confusion matrix on the sampled tweets.

	Label +	Label -
Actual +	209	0
Actual -	91	300

Our goal is to estimate precision and recall of our model on the whole British-Airways dataset, which contains more than 300k unlabeled tweets. To this end, we use the procedure described in Section 4.2. Given that our approach labels approximately 3400 tweets as negative PI, we obtain a 0.95-lower bound of 64.5% on precision and a lower bound of 86.6% on recall for the whole dataset.

4.3.3 Events with largest impact on negative PI. Armed with our negative PI classifier, we initiate a study on which kind of events affect the negative PI the most. Figure 1 focuses on tweets related to British Airways. It plots the cumulative probability value of negative PI ($\sum_{m\in[t-w;t]} \widehat{P}(y_m = 1)$) over time, starting from August 1st 2018 until November 3rd 2018. We observe two main peaks which roughly correspond to two events that hit the airline company in that timeframe: the deplaning of an Indian family because their 3-years-old child was crying (the first peak) and a data breach involving the credit cards of British Airways customers (the second peak). See Table 3 for additional information.

We observe that the first event affects the negative PI of customers more significantly than the second event. A study on why this is the case is beyond the scope of our work. One could speculate that the first event additionally involves an emotional component in potential British Airways customers.

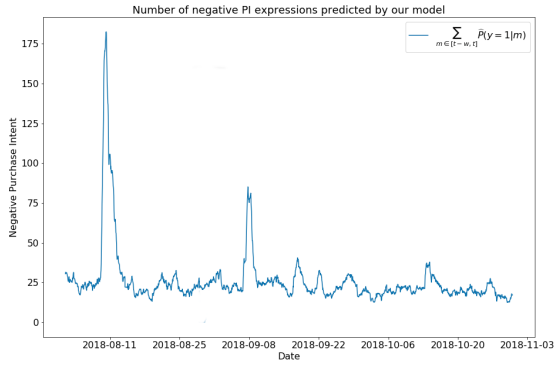

Figure 1: $\sum_{m\in[t-w;t]} \widehat{P}(y_m = 1)$ for all messages m in time window $[t-w;t]$, for window = 24h, step = 1h. The zeros observed sometime after the 25th August is due to issues in data collection.

5 CONCLUSIONS AND FUTURE WORK

We developed a binary classifier for negative purchase intent in posts from social media, such as Twitter. This is the intention of a user of not purchasing any product or consuming any service from a given company. Our problem finds application in brand reputation management, financial investments, and many others. It also allows a study on what kind of events affect the customer purchase intent the most.

We develop a model that exploits the locality of negative PI in social media posts. Our model is a variant of logistic regression which represents a tweet as a set of overlapping "neighborhoods". We perform an extensive experimental evaluation against state-of-the-art approaches on real-world data. In particular, we focus on data collected from Twitter related to scandals which affected British Airways. Our extensive experimental evaluation against state-of-the-art approaches demonstrates the effectiveness of our approach.

For future work, we aim at extending our approach to other domains and continue our study on what kinds of events affect customer purchase intent the most. We shall also put major efforts in improving the performances of our model even further.

REFERENCES

[1] Qi An, Morten Grimmig Christensen, Annith Ramachandran, Raghava Rao Mukkamala, and Ravi Vatrapu. 2018. Volkswagen's Diesel Emission Scandal:

Analysis of Facebook Engagement and Financial Outcomes. In *Big Data - BigData 2018 - 7th International Congress, Held as Part of the Services Conference Federation, SCF 2018, Seattle, WA, USA, June 25-30, 2018, Proceedings*. 260–276.

[2] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*. 4171–4186.

[3] 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, January 25-30, 2015, Austin, Texas, USA*. 2389–2395.

[4] G. Efthimious. 2010. *Regaining Altitude: A case analysis of the JetBlue Airways Valentine's Day 2007 crisis.* The handbook of crisis comm. Handbooks in Comm. and Media. Wiley-Blackwell.

[5] Felix A. Gers, Jürgen Schmidhuber, and Fred A. Cummins. 2000. Learning to Forget: Continual Prediction with LSTM. *Neural Computation* 12, 10 (2000), 2451–2471. https://doi.org/10.1162/089976600300015015

[6] 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, USA, June 1-4, 2014.*

[7] Mohamed Hamroun, Mohamed Salah Gouider, and Lamjed Ben Said. 2016. Large Scale Microblogging Intentions Analysis with Pattern Based Approach. In *Knowledge-Based and Intelligent Information & Engineering Systems: Proceedings of the 20th International Conference KES-2016, York, UK, 5-7 September 2016.* 1249–1257.

[8] Matthew Honnibal. 2015. https://spacy.io/

[9] Clayton J. Hutto and Eric Gilbert. 2014. VADER: A Parsimonious Rule-Based Model for Sentiment Analysis of Social Media Text. In *Proceedings of the Eighth International Conference on Weblogs and Social Media, ICWSM 2014, Ann Arbor, Michigan, USA, June 1-4, 2014.*

[10] Yann LeCun and Yoshua Bengio. 1994. Word-level training of a handwritten word recognizer based on convolutional neural networks. In *12th IAPR International Conference on Pattern Recognition, Conference B: Patern Recognition and Neural Networks, ICPR 1994, Jerusalem, Israel, 9-13 October, 1994, Volume 2*. 88–92.

[11] M. S. Gouider M. Hamroun and L. Ben Said. 2015. *Customer Intentions Analysis of Twitter Based on Semantic Patterns.* WISDOM '15, Sydney.

[12] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. In *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*.

[13] 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 *Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, June 9-14, 2013, Westin Peachtree Plaza Hotel, Atlanta, Georgia, USA*. 380–390.

[14] Jaram Park, Meeyoung Cha, Hoh Kim, and Jaeseung Jeong. 2012. Managing Bad News in Social Media: A Case Study on Domino's Pizza Crisis. In *Proceedings of the Sixth International Conference on Weblogs and Social Media, Dublin, Ireland, June 4-7, 2012.*

[15] Viktor Pekar and Jane M. Binner. 2017. Forecasting Consumer Spending from Purchase Intentions Expressed on Social Media. In *Proceedings of the 8th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis, WASSA@EMNLP 2017, Copenhagen, Denmark, September 8, 2017*. 92–101.

[16] Stefan Stieglitz, Milad Mirbabaie, and Tobias Potthoff. 2018. Crisis Communication on Twitter during a Global Crisis of Volkswagen - The Case of "Dieselgate". In *51st Hawaii International Conference on System Sciences, HICSS 2018, Hilton Waikoloa Village, Hawaii, USA, January 3-6, 2018*. 1–10.

[17] Pedersen T and Banerjee S. 2011. http://search.cpan.org/~tpederse/WordNet-Similarity-2.05/lib/WordNet/stem.pm

[18] Tan Thongtan and Tanasanee Phienthrakul. 2019. Sentiment Classification Using Document Embeddings Trained with Cosine Similarity. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28 - August 2, 2019, Volume 2: Student Research Workshop*. 407–414.

[19] R. Pontes V. Pinheiro and V. Furtado. 2015. A #hashtagtokenizer for Social Media Messages. *Int. J. Comput. Linguistics Appl.* 6, 2 (2015), 141–158. http://www.ijcla.bahripublications.com/2015-2/IJCLA-2015-2-pp-141-158-A-hashtagtokenizer-for-Social-Media-Messages.pdf

[20] Jinpeng Wang, Gao Cong, Wayne Xin Zhao, and Xiaoming Li. 2015. Mining User Intents in Twitter: A Semi-Supervised Approach to Inferring Intent Categories for Tweets. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA*. 318–324.

[21] Juyang Weng, Thomas S. Huang, and Narendra Ahuja. 1992. Object Recognition by a Self-Organizing Neural Network which Grows Adaptively. In *Parallel Image Analysis, Second International Conference, ICPIA '92, Ube, Japan, December 21-23, 1992, Proceedings*. 32–33.

[22] E. B. Wilson. 1927. Probable inference, the law of succession, and statistical inference. *J. Amer. Statist. Assoc.* (1927), 209–212.

[23] M. Hsu M. Castellanos Z. Chen, B. Liu and R. Ghosh. 2013. *Identifying Intentions Posts in Discussion Forums.* Proceedings of NAACL-HLT 2013, pages 1041-1050.