Inferring Advertiser Sentiment in Online Articles using Wikipedia Footnotes

Shaunak Mishra
Yahoo Research
shaunakm@verizonmedia.com

Aasish Pappu*
Spotify Research
aasishp@spotify.com

Narayan Bhamidipati
Yahoo Research
narayanb@verizonmedia.com

ABSTRACT

Online advertising platforms in partnerships with media companies typically have access to an online user's history of viewed articles. If a concerned brand (advertiser) plans to run advertisement campaigns on users exposed to negative articles, it is essential to first identify articles with negative sentiment about the brand. For an advertising platform, scalable identification of such articles with little human-annotation effort is necessary for launching campaigns soon after an advertiser signs up. In this context, generic sentiment analysis tools suffer from the lack of contextual world knowledge associated with the advertiser. Human annotation of articles for supervised approaches is laborious and painstaking. To address these problems, we propose the use of publicly available Wikipedia footnote references for an advertiser, and propagate their sentiment to several articles related to the advertiser. In particular, our proposed approach has three components: (i) automatically find Wikipedia references which have negative sentiment about an advertiser, (ii) learn distributed representations (doc2vec) of article texts referred in footnotes and other unlabeled articles, and (iii) inferring sentiment in unlabeled articles using label propagation (from references) in the doc2vec space. Our experiments spanning three real brands, and data from a major advertising platform (Yahoo Gemini) show significant lifts in sentiment inference compared to existing baselines. In addition, we share valuable insights on how article sentiment influences the online activities of a user with respect to a brand.

CCS CONCEPTS

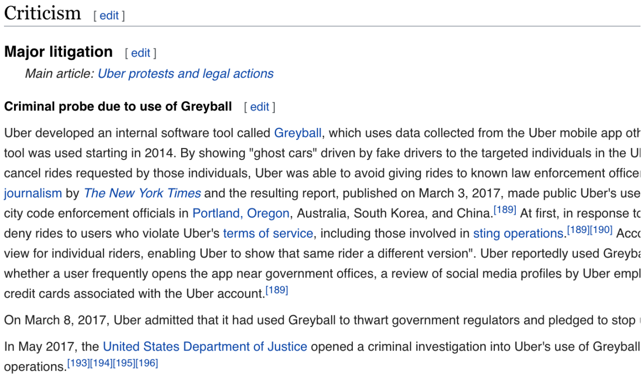
• **Information systems** → *Online advertising*.

ACM Reference Format:
Shaunak Mishra, Aasish Pappu, and Narayan Bhamidipati. 2019. Inferring Advertiser Sentiment in Online Articles using Wikipedia Footnotes. In *Companion Proceedings of the 2019 World Wide Web Conference (WWW '19 Companion), May 13–17, 2019, San Francisco, CA, USA.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3308560.3316752

1 INTRODUCTION

A brand's image is as good as its active users perceive it [7]. It is found that active and loyal users of a brand can influence the attitude of former users and yet-to try users. Thus, when a brand suffers negative press, it may consider advertising policies should aim to keep the loyal users in confidence at the same time persuade non-users to convert to users. The challenge could be that a brand may face a public relations crisis, thereby online users are exposed to critical articles thus it impacts brand's revenue [33]. Such articles with negative sentiment about the brand, may influence users who were loyal to the brand in the past, and even nudge them towards the brand's competitors. Among other means to uplift its image [25] and retain its loyal users, the brand may resort to positive advertising. In addition, to be more effective, the brand (advertiser) can focus on the set of users who were exposed to negative sentiments about it [17].

Major advertising platforms typically have collaborations with (or are a part of) online media companies which publish articles, and track users viewing those articles. When an advertiser is interested in finding a segment of users exposed to negative sentiments, it is essential to first identify the articles with negative sentiment on the advertiser.



Criticism [edit]

Major litigation [edit]

Main article: Uber protests and legal actions

Criminal probe due to use of Greyball [edit]

Uber developed an internal software tool called Greyball, which uses data collected from the Uber mobile app oth
tool was used starting in 2014. By showing "ghost cars" driven by fake drivers to the targeted individuals in the Ul
cancel rides requested by those individuals, Uber was able to avoid giving rides to known law enforcement office
journalism by *The New York Times* and the resulting report, published on March 3, 2017, made public Uber's use
city code enforcement officials in Portland, Oregon, Australia, South Korea, and China.[189] At first, in response tc
deny rides to users who violate Uber's terms of service, including those involved in sting operations.[189][190] Accc
view for individual riders, enabling Uber to show that same rider a different version". Uber reportedly used Greyba
whether a user frequently opens the app near government offices, a review of social media profiles by Uber empl
credit cards associated with the Uber account.[189]

On March 8, 2017, Uber admitted that it had used Greyball to thwart government regulators and pledged to stop

In May 2017, the United States Department of Justice opened a criminal investigation into Uber's use of Greyball
operations.[193][194][195][196]

Figure 1: Screenshot of *criticism* section of the Wikipedia page on Uber. The footnotes mentioned in this section are references to articles with negative sentiment on Uber.

Given a large (web scale) collection of online articles on an advertiser, to automatically predict the sentiment of articles, it is preferable to use a method which is: (i) scalable, (ii) domain (advertiser) oriented (iii) afresh with latest topics associated with advertiser, and (iv) distant supervised. For an advertising platform, such preferences are driven by ad-campaign efficiency and low latency while launching campaigns once an advertiser signs up.

In the context of the preferences mentioned above, existing approaches for sentiment analysis are open-domain or trained on generic datasets e.g., Yelp or IMDB. They may be ineffective in a niche domain such as texts about an advertiser. Labeled data

[*] Work done while the author was at Yahoo Research.

WWW '19 Companion, May 13–17, 2019, San Francisco, CA, USA

ACM ISBN 978-1-4503-6675-5/19/05.
https://doi.org/10.1145/3308560.3316752

for a niche domain is rare, and expensive to obtain. Most existing approaches for sentiment classification are either supervised models or use transfer learning, hence they are not viable. Furthermore, to keep up with new topics, the annotation task needs to be done on a regular basis on articles with such new topics for every advertiser's domain. Given such shortcomings, we explore a distant-supervised approach that leverages Wikipedia to predict sentiment on unlabeled news articles of an advertiser. A Wikipedia page on an advertiser typically has a section with negative sentiment (*e.g.*, the criticism section), and the footnotes (references to other online articles) mentioned in such a section are human-curated examples of articles with negative sentiment about the advertiser (illustrated in Figure 1. Such footnotes not only provide labeled examples of negative articles about the advertiser, but also keep up with current topics since Wikipedia articles on major entities get updated in a matter of minutes. Footnotes from other (non-criticism) sections in the Wikipedia page on the advertiser can serve as examples of positive (or neutral) articles on the advertiser.

In this paper, we propose an approach which automatically collects the text in Wikipedia footnote articles for an advertiser, and labels them negative or positive depending on the section they are referred in. For the task of inferring sentiment in unlabeled news articles on an advertiser, we first obtain doc2vec [19] embeddings of the footnote articles and the unlabeled articles, and then propagate sentiment from the (labeled) footnote articles to the unlabeled articles. Our main contributions can be summarized as follows:

(1) a scalable sentiment analysis method using Wikipedia footnotes which achieves significant lifts in average precision (AP) and area under ROC curve (AUC) compared to baselines (as high as 19% lift in AP for an advertiser),
(2) an analysis of how the (predicted) sentiment in articles affects online interactions (ad clicks and purchases) of users with the advertiser. We quantitatively validate the intuitive expectation that users exposed to positive sentiments tend to have a higher click through rate (CTR) and conversion rate (CVR) for ads corresponding to the advertiser. In addition, we show that the sentiment for a particular advertiser may or may not carry over to the advertiser's competitors in terms of ad interactions.

2 RELATED WORK

In this section, we cover prior work related to sentiment analysis tools, brand specific sentiment inference, and user behavior models in online advertising.

2.1 Sentiment analysis tools

Sentiment analysis has a huge body of literature, and many state-of-the-art methods are now readily available as tools for text classification; this includes (i) Senticnet [8, 9], (ii) Stanford's CORENLP [2, 31], (iii) NLTK-SentiWordNet [1, 5], (iv) TextBlob [3, 23], (v) VADER [4, 15], and (vi) Polyglot [10]. In this paper, the six tools listed above serve as baselines for comparison against our proposed methods, and additional details regarding each of these methods are described in Section 4.2. Apart from the methods listed above, there has been recent work focusing on review data sets (*i.e.*, IMDB reviews and Amazon reviews) using (supervised) deep learning methods [21, 26, 28, 32, 36]. We do not consider such methods in our paper as we focus on: (i) brand specific sentiment in online news articles, and (ii) unsupervised and semi-supervised methods for inferring sentiment.

2.2 Brand specific sentiment

In terms of inferring sentiment specific to a brand, existing work broadly spans two classes: (i) sentiment prediction using Twittter data (*i.e.*, tweets on a brand) [11, 14, 18, 34], and (ii) investor sentiment towards a brand leading to stock price movement [13, 20, 30]. Specific details for prior work in both the classes are provided below; we first go over prior work using Twitter, followed by work on investor sentiment.

2.2.1 Sentiment analysis using Twitter data.

In [14], a supervised approach was proposed for predicting consumer sentiment towards a brand based on tweets. The approach involved constructing a Twitter-specific sentiment lexicon (including tokens related to a brand), which was used to create feature vectors fed to a support vector machine (SVM). In [34], a similar approach was used to extract brand specific features from a smaller set of tweets for mobile companies in Indonesia. The authors measured customer satisfaction for five products for each of the three brands considered in the paper. In [18], 150K tweets were analyzed for linguistic structure and key phrases that convey brand sentiment. The authors also analyzed corporate accounts of specific brands for frequency, timing, and content of their tweets. In [11], tweets were analyzed based on a fixed lexicon of emotional words. The aggregate sentiment of emotional words [27] in a tweet was visualized for corporate users and used to monitor evolving sentiment for brand-specific events.

2.2.2 Investor sentiment and stock prices.

Financial news is an invaluable resource to study investors' sentiment towards stocks, and their correlation with stock price movements. Most works described below study the correlation between a public company's evolving sentiment, and their stock price movement. In [30], an active learning approach was introduced to forecast stock price movements via sentiment analysis of stock-related tweets. The authors used an SVM classifier to predict sentiment, and Granger causality test was used to validate stock sentiment as an indicator for its price movements. In [13], a metric based on lexical cohesion was proposed; this measured the sentiment, intensity, and polarity of text. This metric was shown to have strong correlation with human judgments in finance news. In [20], the authors built a stock price prediction framework using the Harvard psychological dictionary and Loughran-McDonald finance sentiment dictionary (for feature engineering). News documents were represented as feature vectors, where the features were tokens in the above lexicons.

In spirit, our focus in this paper is similar to the works described above: inferring brand (advertiser) sentiment in online news articles, and studying their effect on online advertising. The major difference our work introduces is the use of Wikipedia (text in footnote articles in particular), which provides a small set of labeled examples enabling unsupervised and semi-supervised methods for sentiment classification. To the best of our knowledge, there has been no prior work on using Wikipedia footnotes towards brand sentiment classification. In addition, we also study the influence of the inferred sentiment on users in the context of online advertising (background given below).

2.3 Online advertising and brand sentiment

In a standard online advertising setup, predictive models are employed to predict user behavior (ad clicks and conversions/purchases) when a particular ad is shown to the user; such models are used to select relevant ads as well as for optimizing revenue from ads [6, 24]. In particular, CTR prediction models predict the chances of a click (for a user-ad pair) given available user, ad, and context features. In a similar spirit, conversion models predict the chances of a user converting or purchasing from a particular brand (advertiser) after exposure the brand's ad. In this paper, we explore impact of user sentiment towards a brand (computed as an aggregate of sentiments in relevant news articles read by the user) on the ad click and conversion behavior. In theory, such brand-user sentiment features can be readily consumed by CTR and conversion prediction models used in the advertising industry (which can be as simple as logistic regression [6, 24], or more complex deep neural networks as in [16, 29]). Again, to the best of our knowledge, there is no prior work quantizing the effect of news article sentiments on ad click and conversion behavior with respect to a brand (or its competitors in the same product category).

3 PROPOSED METHOD

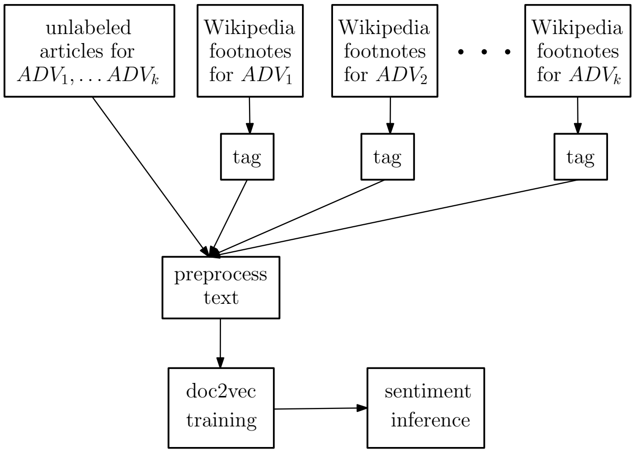

unlabeled articles for $ADV_1, \ldots ADV_k$
Wikipedia footnotes for ADV_1
Wikipedia footnotes for ADV_2
. . .
Wikipedia footnotes for ADV_k
tag
tag
tag
preprocess text
doc2vec training
sentiment inference

Figure 2: Overview of the proposed architecture.

In this section, we first describe the setup formally in Section 3.1. This is followed by Section 3.2 which gives an overview of the proposed architecture. Section 3.3 covers doc2vec training to obtain embeddings (of labeled footnote articles and unlabeled news articles), and Section 3.4 describes the details of sentiment inference from the doc2vec embeddings.

3.1 Setup

We consider a setup with k advertisers: $ADV_1,\ ADV_2, \ldots,\ ADV_k$. For each advertiser ADV_i there is a corresponding Wikipedia page $Wiki_i$. In $Wiki_i$, $foot_{i,neg}$ is set of footnote articles with negative label (*i.e.*, label = -1) and $foot_{i,pos}$ is the set of articles with positive or neutral label (*i.e.*, label = 1). The labels are assigned based on the section in which the footnotes are mentioned, *i.e.*, only the footnotes in criticism related sections are marked negative[1]. Given such labeled footnote articles (*i.e.*, the entire text in those articles rather than just headlines), we focus on predicting the advertiser sentiment for an unlabeled article $u_{i,j} \in U_i$ where U_i is the set of unlabeled articles for advertiser ADV_i.

3.2 Architecture

Figure 2 shows the proposed architecture. The data collection step includes collecting all Wikipedia footnote articles for all k advertisers. The tag step (labeling) for footnote articles is done on the basis of the section the footnote reference is made (*i.e.*, articles referred to in the *criticism* section of the Wikipedia page are automatically labeled negative; illustrated in Figure 1). The preprocess block cleans the text in the articles (filters out non-English documents, removes punctuation) prior to the doc2vec training step.

The doc2vec block takes as input the *cleaned* text in all labeled footnote articles (across all advertisers) as well as unlabeled articles (details in Section 3.3 below), and produces low dimensional embeddings for each article. Finally using the doc2vec embeddings, sentiment inference is done using the algorithms described in Section 3.4.

3.3 Doc2vec training

We obtain low dimensional embeddings of all articles (labeled footnotes and unlabeled articles) across all advertisers using doc2vec [19]. Doc2Vec learns low dimensional embeddings for words and documents (articles) from a large corpus in an unsupervised manner. In particular, doc2vec embeddings of articles with high semantic similarity tend to be *close* to each other in terms of cosine similarity between the learnt embeddings. In our experiments, we use the vanilla version of doc2vec [19], as well as an enhanced version as proposed below.

3.3.1 Opinion observer enhanced doc2vec (OO+). We use the vocabulary in the seminal *Opinion Observer* work [22] to obtain a set of negative words (denoted by set N_{OO}), and enhance doc2vec training in the following manner. For each negative sentiment footnote article, we identify words/phrases which are also present in N_{OO}, and replace them in the article text by *criticism* tag *e.g.*, *pricey* is replaced by *criticism_pricey*. By doing so, we establish a stronger connection across all negative footnote articles prior to the doc2vec training step. In the remainder of this paper, we will refer to this as the OO+ version of doc2vec (as opposed to the vanilla version described above).

3.4 Sentiment inference

Using the doc2vec embeddings of labeled footnote articles and unlabeled articles on advertisers, we propagate the sentiment from labeled articles (referred in footnotes under *criticism* section) using the two approaches described below.

[1] There is a small degree of label noise in the footnote labeling procedure based on Wikipedia sections; there are cases when a footnote mentioned in a non-criticism section is actually of negative sentiment. In this paper, we do not consider such label noise, but evaluate our proposed methods on an independently annotated data set.

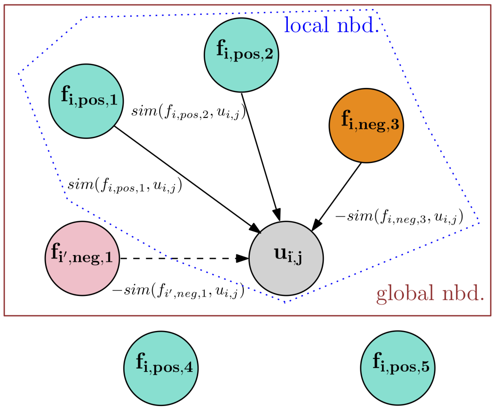


Figure 3: Nearest neighbour sentiment inference in the doc2vec embedding space. In the local mode, the average of the footnote labels of the same advertiser (weighted by cosine similarity) is considered, while in the global mode footnotes across all advertisers are considered.

3.4.1 Nearest neighbour. In the nearest neighbour approach, we simply obtain the average sentiment score of an unlabeled article by a weighted average of the labels in its neighbourhood. The cosine similarity between the doc2vec embedding of the unlabeled article and a labeled article in its neighbourhood is used as the weight during the averaging process. The nearest neighbour approach has multiple parameters which can be used for tuning the approach for each advertiser.

(1) We consider only the top m_i neighbours (ranked by cosine similarity) while doing the weighted average for each unlabeled article for ADV_i.
(2) For an unlabeled article on ADV_i, we can either consider only ADV_i related footnotes (*i.e.*, local mode) or consider all footnotes across multiple advertisers (*i.e.*, global mode) as illustrated in Figure 3.

Intuitively, the global mode may be more effective when the number of footnotes for an advertiser is low.

3.4.2 Label propagation. Label propagation is a family of semi-supervised algorithms based on graph representations. Both labeled and unlabeled instances in the data are considered as vertices, and a (vertex) similarity function is used to assign edge weights between a pair of vertices. At a high level, label propagation algorithms exploit the manifold structure in unlabeled data, and assign (propagate) labels from labeled vertices to unlabeled vertices. In the past, several text-based problems have been formulated in this setting [12, 35, 35]. In this paper, we use the label propagation method proposed in [37], and construct an affinity matrix of document representations (doc2vec vectors) for both labeled and unlabeled news articles in our data set. In particular, we use cosine similarity between the doc2vec vectors as the edge weight between a pair of vertices. The label propagation algorithm iteratively assigns labels in high-density areas of unlabeled documents in the doc2vec space; it also learns parameters using minimum spanning tree heuristic, and entropy minimization to fine-tune label assignment. In a spirit similar to global and local modes in the nearest neighbour approach, we use the term global mode for label propagation when we consider a graph with unlabeled articles and footnote articles across all advertisers. In the local mode, we consider a graph with unlabeled articles and footnote articles only for the concerned advertiser.

adv	# wiki	# +ve	#	# labeled	# +ve
	notes	wiki	articles	articles	labeled
ADV_1	221	125	6921	284	83
ADV_2	82	48	1479	205	150
ADV_3	40	27	4314	257	168

Table 1: Articles data for each advertiser.

4 RESULTS

In this section, we describe results on the accuracy of the proposed sentiment analysis methods, as well as associated user behaviour insights. We first describe our data sources in Section 4.1, followed by Section 4.2 on baseline methods, and Section 4.3 on their performance in comparison to the proposed methods. This is followed by Section 4.4 on insights obtained from user interactions with ads after reading articles spanning diverse sentiments on various brands (advertisers).

4.1 Data

We conducted experiments on a mix of public (Wikipedia) and proprietary data for three advertisers (anonymized as ADV_1, ADV_2, ADV_3). In particular, ADV_1 is an employer for drivers in the transportation domain, ADV_2 is an e-commerce portal, and ADV_3 is a wireless (phone) service provider. The data was collected as described below, and the relevant counts are summarized in Table 1.

4.1.1 Wikipedia data. We collected data on ADV_1, ADV_2, and ADV_3 from Wikipedia (*i.e.*, the text from online articles listed as footnotes in a brand's Wikipedia page). A footnote article associated with the criticism section of a brand's Wikipedia page was marked as having negative sentiment, and the rest were marked positive. The number of footnote articles per advertiser is shown in Table 1 along with the count of footnotes marked positive (or neutral). In addition, a visualization of the doc2vec embeddings of the footnote articles reveals a sense of separability between positive and negative footnote articles as shown in Figure 4.

4.1.2 Data from Yahoo! articles. We collected online articles related to ADV_1, ADV_2, and ADV_3 which appeared in Yahoo Finance, Yahoo Sports, and Yahoo News during the period September 2017 - October 2018. From this collection of online articles, an editorial team selected articles (with higher counts of users who read them) and annotated them with sentiments. The selected articles were primarily about one of the three advertisers in consideration. The count of annotated (labeled) articles per advertiser is summarized in Table 1. For example, for ADV_2 there were 1479 articles, and out of those only 205 were labeled. In addition, we obtained anonymized data from a major advertising platform (Yahoo Gemini) regarding trails of online user activities, *e.g.*, anonymized history of a user's

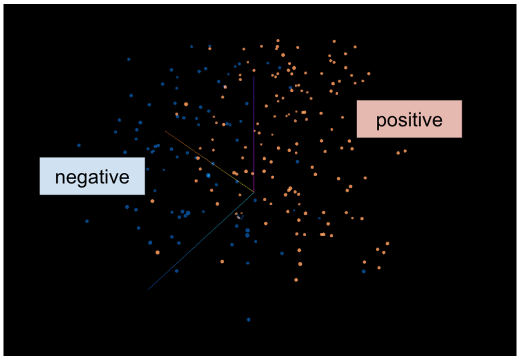


Figure 4: Visualization of 100 dimensional doc2vec embeddings for ADV_1 footnote articles using TensorFlow embedding projector. The blue dots represent footnote articles with negative sentiment, and the orange dots represent footnote articles with positive sentiment.

article views, ad clicks and conversions (*e.g.*, purchases, sign-ups, installs).

4.2 Baseline sentiment analysis methods

To benchmark our results against existing state-of-the-art sentiment methods, we ran six sentiment classifiers on our testing data. We choose four lexicon or knowledge-base based methods and two statistical methods as our baselines.

4.2.1 Senticnet. Senticnet [8, 9] is a semi-automatically constructed sentiment resource using semantic web techniques to glean opinions from natural language. It adopts an energy-based formalism as used in COGBASE to connect multi-word expressions with semantic concepts. Previous versions of Senticnet used an ensemble of graph-mining, and dimensionality reduction algorithms.

4.2.2 Stanford NLP. Stanford's CORENLP provides a sentiment analysis classifier that is based on compositional model over trees using deep learning [2, 31]. The classifier assigns a sentiment score to a sentence in a document where every sentence is represented as a binary tree.

4.2.3 NLTK-SentiWordNet. NLTK-SentiWordNet [1, 5] extends WordNet with numerical annotations for every synset in a collection of 147,306 synsets. These numerical annotations correspond to positive, negative and neutral classes.

4.2.4 TextBlob. TextBlob is a text processing toolkit that includes sentiment classifier as one of its many text analysis tools [3, 23]. It uses a semi-automatically constructed subjectivity lexicon of adjectives, where each adjective has a polarity score (negative is -1.0, positive is +1.0) and a subjectivity score (objective: 0.0 to subjective 1.0). These scores are further annotated for reliability of the assignment (1.0 for human vs 0.7 for automatic).

advertiser	method	avg precision	AUC
ADV_1	TextBlob	0.295	0.559
ADV_1	VADER	**0.577**	**0.784**
ADV_1	SentiWordNet	0.446	0.692
ADV_1	SenticNet	0.398	0.623
ADV_1	Polyglot	0.462	0.702
ADV_1	Stanford	0.357	0.623
ADV_2	TextBlob	**0.855**	**0.734**
ADV_2	VADER	0.832	0.690
ADV_2	SentiWordNet	0.813	0.633
ADV_2	SenticNet	0.784	0.589
ADV_2	Polyglot	0.832	0.678
ADV_2	Stanford	0.853	0.689
ADV_3	TextBlob	0.748	0.617
ADV_3	VADER	0.798	0.661
ADV_3	SentiWordNet	0.776	0.665
ADV_3	SenticNet	0.721	0.584
ADV_3	Polyglot	**0.803**	**0.677**
ADV_3	Stanford	0.697	0.579

Table 2: Baseline results for each advertiser.

4.2.5 Vader Sentiment. VADER is Valence Aware Dictionary for Sentiment Reasoning tool. It is a rule-based method that empirically bootstraps a lexicon of tokens [4, 15]. The rules are primarily grammatical and syntactical in nature and are aimed to extract intensity of words that convey sentiment. The authors of VADER report state-of-the-art results that match [31] results on various benchmark data sets include Amazon reviews corpus and IMDB review data set.

4.2.6 Polyglot. Polyglot is a multilingual natural language toolkit that includes sentiment classification as one of the many tools[10]. The classifier is based on graph-propagation that connects most frequently used words across 136 languages on Wikipedia. The links/connections between the words help propagate the sentiment across various languages using graph propagation and label propagation methods.

4.3 Evaluation on labeled data

4.3.1 Baselines: Table 2 shows the performance of the six baselines (listed in Section 4.2) on the labeled data for each advertiser. In particular, to obtain the sentiment *score* of each article, we averaged the sentiment score (given per line by the baseline method) across all lines in the article. The performance is measured in terms of average precision and area under the ROC curve (AUC). As shown in Table 2, VADER, TextBlob and Polyglot are the best baseline methods for for ADV_1, ADV_2 and ADV_3 respectively.

4.3.2 Proposed approaches. To compare with the best baseline results for each advertiser, Table 3 shows the evaluation results for the proposed methods (the first row for each advertiser corresponds to the best baseline for the advertiser). As shown in Table 3, for ADV_1, the label propagation method with vanilla doc2vec embeddings, and propagation using only ADV_1 footnote articles (*i.e.*, local mode) has the best performance. A plausible reason behind its success could

advertiser	method	doc2vec version	neighborhood	avg precision	AUC
ADV_1	VADER	-	-	0.577	0.784
ADV_1	label prop	OO+	global	0.596	0.844
ADV_1	label prop	OO+	local	0.642	0.865
ADV_1	label prop	vanilla	global	0.618	0.851
ADV_1	label prop	vanilla	local	**0.685**	**0.876**
ADV_1	nearest neighbor	OO+	global	0.613	0.830
ADV_1	nearest neighbor	OO+	local	0.616	0.834
ADV_1	nearest neighbor	vanilla	global	0.614	0.825
ADV_1	nearest neighbor	vanilla	local	0.628	0.823
ADV_2	TextBlob	-	-	**0.855**	**0.734**
ADV_2	label prop	OO+	global	0.800	0.599
ADV_2	label prop	OO+	local	0.779	0.565
ADV_2	label prop	vanilla	global	0.820	0.622
ADV_2	label prop	vanilla	local	0.758	0.508
ADV_2	nearest neighbor	OO+	global	0.783	0.586
ADV_2	nearest neighbor	OO+	local	0.793	0.585
ADV_2	nearest neighbor	vanilla	global	0.797	0.595
ADV_2	nearest neighbor	vanilla	local	0.799	0.586
ADV_3	Polyglot	-	-	0.803	0.677
ADV_3	label prop	OO+	global	0.862	0.803
ADV_3	label prop	OO+	local	0.768	0.668
ADV_3	label prop	vanilla	global	0.851	0.783
ADV_3	label prop	vanilla	local	0.715	0.613
ADV_3	nearest neighbor	OO+	global	**0.878**	**0.808**
ADV_3	nearest neighbor	OO+	local	0.720	0.624
ADV_3	nearest neighbor	vanilla	global	0.861	0.792
ADV_3	nearest neighbor	vanilla	local	0.702	0.596

Table 3: Performance of the proposed methods for each advertiser.

be the presence of a large number of available footnote articles (*i.e.*, 221) for ADV_1. In contrast, for ADV_3, which has a very few footnote articles (*i.e.*, 40), the nearest neighbor classifier using global mode (*i.e.*, using footnotes for all the three advertisers) has the best performance. In the case of ADV_2, none of the proposed approaches are better than the best baseline (TextBlob); however, label propagation in the global mode comes closest in performance to TextBlob[2]. The results also indicate that the nearest neighbour approaches are usually competitive in performance to the label propagation approaches. This is encouraging on the scalability front since there are very efficient methods (*e.g.*, locality sensitive hashing) for looking up neighboring footnote articles in the doc2vec space.

4.4 User Behaviour Insights

On grounds of scalability, we used the best nearest neighbor approach for each of the three advertisers, and produced sentiment scores for each article associated with an advertiser (*i.e.*, 6921 articles for ADV_1, 1479 for ADV_2, and 4314 for ADV_3). For each advertiser, we identified online users (using data from the advertising platform) who had read online articles about the advertiser in the September 2017 - October 2018 time window (*i.e.*, exposed users). We also identified if the set of exposed users had (i) clicked on the advertiser's ads, (ii) clicked on the advertiser's competitor's ads, and (iii) converted on the advertiser (e.g., purchase from the advertiser). Using the above data, we inferred the (advertiser specific) user sentiment prior to a target event (*i.e.*, advertiser ad click or advertiser conversion or competitor ad click) in the following manner:

$$\texttt{sentiment}_{user,ADV} = \frac{\sum_{\text{article}\in\text{history}} \texttt{sentiment}_{article}}{|\text{articles} \in \text{history}|},$$

where *history* is the set of articles on *ADV* read by the user before the target event, and $sentiment_{article}$ is the sentiment score of the article as obtained by the nearest neighbour approach. If the user did not do the target event in the September 2017 - October 2018 time window, the history includes all articles read by the user on the advertiser. Using the above definition of user sentiment for each advertiser, we obtained the user sentiment across all exposed users and divided them into two groups: positive users and negative users (by using a threshold on the user sentiment score). Table 4 shows the target event rate for positive and negative users (rate normalized by the average target event rate for the entire set of exposed users); the target events in the table include: (i) conversion (*i.e.*, if the user purchased/converted after exposure), (ii) ad click (*i.e.*, if the user clicked on the advertiser's ad after exposure), and (iii) competitor ad click (*i.e.*, if the user clicked on a chosen competitor's ad after exposure to the advertiser's articles). For example, as shown

[2] On further examination, we found that the labeled footnotes for ADV_2 had a small degree of label noise, which could be causing inferior performance using our proposed method.

adv.	# exposed in mil.	target event	target event rate	+ve user target rate lift	-ve user target rate lift
ADV_1	2.17	conversion	0.0089	1.0637	0.8749
ADV_1	2.17	ad click competitor	0.1081	1.0400	0.9582
ADV_1	2.17	ad click	0.0676	0.9748	1.0261
ADV_2	0.97	conversion	0.0836	1.0228	0.8817
ADV_2	0.97	ad click competitor	0.1020	1.0274	0.8737
ADV_2	0.97	ad click	0.4821	1.0116	0.9444
ADV_3	1.96	conversion	0.0369	1.0903	0.8915
ADV_3	1.96	ad click competitor	0.1098	1.0116	0.9862
ADV_3	1.96	ad click	0.0258	0.9959	1.0025

Table 4: Impact of user sentiment on ad clicks and conversions.

in Table 4, there were about 0.97 million exposed users for ADV_2, and out of them the positive users had a normalized conversion rate of 1.0228 while the negative users had a normalized conversion rate of 0.8817. Based on this, the positive users were 16% more likely to convert on ADV_2 compared to negative users. The patterns for ad clicks and conversions with regards to user sentiment are similar across all advertisers; however, when it comes to behaviour towards ads of competitors, there is a difference. For ADV_2, users with positive sentiment are more likely to click on ads of the competitor (*i.e.*, another e-commerce company) than negative users. However, for ADV_3, negative users for ADV_3 are more likely to click on a competitor's ads (*i.e.*, another wireless service provider). These insights are valuable towards consuming sentiment signals in click and conversion models, and serve as useful inputs the advertiser as well.

5 CONCLUSIONS AND FUTURE WORK

In this paper, we explore the use of Wikipedia footnote articles towards advertiser specific sentiment analysis. Using a small number of foot note articles (~ 200), our approach was able to outperform six competitive baselines for two out of three advertisers in our experiments. The ability to propagate sentiments to a large number of unlabeled news articles on an advertiser, enables us to not only identify users exposed to negative sentiments, but also quantify the impact on ad clicks and conversions. However, as discussed below, a few challenging topics remain open and are directions for future work.

(1) During the data validation phase (via random checks on the foot note labels), we identified Wikipedia footnotes from the *criticism* section which were neutral. We also noticed articles in the non-criticism section which were of negative sentiment. Although small in number, such inconsistencies introduce label noise in our setup. Quantifying such label noise across advertisers (with minimum human involvement), and making our sentiment propagation approach more robust are directions for future work.

(2) In our experiments, we found that for an advertiser with negligible number of footnote articles, it is better to propagate sentiments from footnotes across multiple advertisers (*i.e.*, the global mode in the nearest neighbour approach). Refining this along the lines of transfer learning is another direction for future work.

REFERENCES

[1] NLTK SentiWordNet. http://www.nltk.org/howto/sentiwordnet.html.

[2] Stanford Core NLP. https://github.com/dasmith/stanford-corenlp-python.

[3] TextBlob. https://github.com/sloria/textblob.

[4] Vader sentiment. https://github.com/cjhutto/vaderSentiment.

[5] S. Baccianella, A. Esuli, and F. Sebastiani. Sentiwordnet 3.0: an enhanced lexical resource for sentiment analysis and opinion mining. In *Lrec*, volume 10, pages 2200–2204, 2010.

[6] N. Bhamidipati, R. Kant, S. Mishra, and M. Zhu. A large scale prediction engine for app install clicks and conversions. In *Proceedings of the 26th ACM International Conference on Information and Knowledge Management (CIKM 2017)*.

[7] M. Bird, C. Channon, and A. S. Ehrenberg. Brand image and brand usage. *Journal of Marketing Research*, pages 307–314, 1970.

[8] E. Cambria, D. Olsher, and D. Rajagopal. Senticnet 3: a common and common-sense knowledge base for cognition-driven sentiment analysis. In *Twenty-eighth AAAI conference on artificial intelligence*, 2014.

[9] E. Cambria, S. Poria, D. Hazarika, and K. Kwok. Senticnet 5: Discovering conceptual primitives for sentiment analysis by means of context embeddings. In *AAAI*. AAAI Press, 2018.

[10] Y. Chen and S. Skiena. Building sentiment lexicons for all major languages. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Short Papers)*, pages 383–389, 2014.

[11] E. Colleoni, A. Arvidsson, L. K. Hansen, and A. Marchesini. Measuring corporate reputation using sentiment analysis. In *Proceedings of the 15th International Conference on Corporate Reputation: Navigating the Reputation Economy, New Orleans, USA*, 2011.

[12] B. Dalvi, E. Minkov, P. P. Talukdar, and W. W. Cohen. Automatic gloss finding for a knowledge base using ontological constraints. In *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining*, pages 369–378. ACM, 2015.

[13] A. Devitt and K. Ahmad. Sentiment polarity identification in financial news: A cohesion-based approach. In *Proceedings of the 45th annual meeting of the association of computational linguistics*, pages 984–991, 2007.

[14] M. Ghiassi, J. Skinner, and D. Zimbra. Twitter brand sentiment analysis: A hybrid system using n-gram analysis and dynamic artificial neural network. *Expert Systems with applications*, 40(16):6266–6282, 2013.

[15] C. H. E. Gilbert. Vader: A parsimonious rule-based model for sentiment analysis of social media text. In *Eighth International Conference on Weblogs and Social Media (ICWSM-14). Available at (20/04/16) http://comp. social. gatech. edu/papers/icwsm14. vader. hutto. pdf*, 2014.

[16] D. Gligorijevic, J. Gligorijevic, A. Raghuveer, M. Grbovic, and Z. Obradovic. Modeling mobile user actions for purchase recommendation using deep memory networks. SIGIR, 2018.

[17] S. J. Hoch and J. Deighton. Managing what consumers learn from experience. *The Journal of Marketing*, pages 1–20, 1989.

[18] B. J. Jansen, M. Zhang, K. Sobel, and A. Chowdury. Twitter power: Tweets as electronic word of mouth. *Journal of the American society for information science and technology*, 60(11):2169–2188, 2009.

[19] Q. Le and T. Mikolov. Distributed representations of sentences and documents. In *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32*, ICML'14, 2014.

[20] X. Li, H. Xie, L. Chen, J. Wang, and X. Deng. News impact on stock price return via sentiment analysis. *Knowledge-Based Systems*, 69:14–23, 2014.

[21] Z. Lin, M. Feng, C. N. d. Santos, M. Yu, B. Xiang, B. Zhou, and Y. Bengio. A structured self-attentive sentence embedding. *arXiv preprint arXiv:1703.03130*, 2017.

[22] B. Liu, M. Hu, and J. Cheng. Opinion observer: Analyzing and comparing opinions on the web. In *Proceedings of the 14th International Conference on World Wide Web*, WWW '05.

[23] S. Loria, P. Keen, M. Honnibal, R. Yankovsky, D. Karesh, E. Dempsey, et al. Textblob: simplified text processing. *Secondary TextBlob: Simplified Text Processing*, 2014.

[24] H. B. McMahan, G. Holt, D. Sculley, M. Young, D. Ebner, J. Grady, L. Nie, T. Phillips, E. Davydov, D. Golovin, S. Chikkerur, D. Liu, M. Wattenberg, A. M. Hrafnkelsson, T. Boulos, and J. Kubica. Ad click prediction: a view from the trenches. KDD 2013.

[25] S. J. Milberg, C. Whan Park, and M. S. McCarthy. Managing negative feedback effects associated with brand extensions: The impact of alternative branding

strategies. *Journal of Consumer Psychology*, 6(2):119–140, 1997.

[26] T. Munkhdalai and H. Yu. Neural semantic encoders. In *Proceedings of the conference. Association for Computational Linguistics. Meeting*, volume 1, page 397. NIH Public Access, 2017.

[27] F. Å. Nielsen. A new anew: Evaluation of a word list for sentiment analysis in microblogs. *arXiv preprint arXiv:1103.2903*, 2011.

[28] A. Radford, R. Jozefowicz, and I. Sutskever. Learning to generate reviews and discovering sentiment. *arXiv preprint arXiv:1704.01444*, 2017.

[29] Y. Shan, T. R. Hoens, J. Jiao, H. Wang, D. Yu, and J. C. Mao. Deep crossing: Web-scale modeling without manually crafted combinatorial features. KDD 2016.

[30] J. Smailović, M. Grčar, N. Lavrač, and M. Žnidaršič. Stream-based active learning for sentiment analysis in the financial domain. *Information sciences*, 285:181–203, 2014.

[31] R. Socher, A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Ng, and C. Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642, 2013.

[32] K. S. Tai, R. Socher, and C. D. Manning. Improved semantic representations from tree-structured long short-term memory networks. *arXiv preprint arXiv:1503.00075*, 2015.

[33] C. C. Verschoor. Uber culture causes big losses: harassment and mismanagement have led to steep losses for this high-flying company. *Strategic Finance*, 99(3):23–25, 2017.

[34] N. A. Vidya, M. I. Fanany, and I. Budi. Twitter sentiment to analyze net brand reputation of mobile phone providers. *Procedia Computer Science*, 72:519–526, 2015.

[35] M. Yu, S. Wang, C. Zhu, and T. Zhao. Semi-supervised learning for word sense disambiguation using parallel corpora. In *Fuzzy Systems and Knowledge Discovery (FSKD), 2011 Eighth International Conference on*, volume 3, pages 1490–1494. IEEE, 2011.

[36] Z.-H. Zhou and J. Feng. Deep forest: Towards an alternative to deep neural networks. *arXiv preprint arXiv:1702.08835*, 2017.

[37] X. Zhu and Z. Ghahramani. Learning from labeled and unlabeled data with label propagation.