APNEA: Intelligent Ad-Bidding Using Sentiment Analysis

Samuel Suraj Bushi
bushi@ualberta.ca
University of Alberta
Edmonton, Alberta, Canada

Osmar R. Zaïane
zaiane@ualberta.ca
University of Alberta
Edmonton, Alberta, Canada

ABSTRACT

Online advertising is one of the most lucrative forms of advertising, making it an important channel of advertising media. Contextual Advertising is a type of online display advertising that takes cues from the content of the triggering page and displays advertisements that are relevant to the current context. However, on several occasions, the *context* may have a negative connotation, and displaying advertisements that are relevant to it might prove to be detrimental to the advertiser. We refer to such a scenario as an *unfortunate placement*. In this work, we propose APNEA (Ad Positive NEgative Analysis), a light-weight system that uses a sentiment-oriented approach to rank the advertisers such that positively correlated brands are ranked higher than brands that are neutral or negatively correlated. Experiments show that APNEA helps avoid unfortunate placements while maintaining ad-relevance. It outperforms several baselines in terms of accuracy on human-annotated test data while having a lower run-time, which is crucial for real-time bidding systems.

CCS CONCEPTS

• **Information systems → Computational advertising**; **Content match advertising**; **Display advertising**; **Sentiment analysis**.

KEYWORDS

advertising, sentiment analysis, contextual advertising

ACM Reference Format:
Samuel Suraj Bushi and Osmar R. Zaïane. 2019. APNEA: Intelligent Ad-Bidding Using Sentiment Analysis. In *IEEE/WIC/ACM International Conference on Web Intelligence (WI '19), October 14–17, 2019, Thessaloniki, Greece.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3350546.3352503

1 INTRODUCTION

Online advertising is one of the most lucrative forms of advertising. In 2017 Q1, more than 85% of Google's revenue came from Online Advertising operations [15]. Other search engines and web-hosting platforms also leverage the monetary benefits of online advertising to a similar extent.

WI '19, October 14–17, 2019, Thessaloniki, Greece

ACM ISBN 978-1-4503-6934-3/19/10…$15.00
https://doi.org/10.1145/3350546.3352503

There are many different forms of online advertising, i.e. text, image, audio, video, email (spam) etc. According to the authors of [3], textual advertisements make up a large part of the market. There are two main channels through which textual advertisements are triggered:

(1) *Sponsored Search:* Sponsored search works by using a user's query in a search engine to trigger ads, which are then displayed as a part of the results. This works by understanding the user's information need and matching it with competing advertisers who have a similar advertising agenda. Most of the popular search engines like Google, Bing, Yahoo!, etc. all use sponsored search on their search platforms.
(2) *Contextual Advertising:* Contextual Advertising, on the other hand, works based on the content of a web page (also known as a *triggering page*) that a user visits with possibly a combination with the available profile of the user, and displays advertisements that are relevant to the *context* of that web page. Several studies have shown that increased relevance indeed improves the click-through-rate of advertisements. [5, 24].

The space for Real Time Bidding (RTB) for display ads is comprised of advertisers constantly bidding with a Demand Side platform (DSP), and suppliers selling ad placements in a Supply Side Platform (SSP). The goal of a DSP is for an ad inventory to buy an audience as cheap as possible via good placements in websites, and the purpose of an SSP is for publishers to sell places for ad banners on their sites the more expensive possible. An ad exchange connects advertisers and publishers via auctions. Through these auctions publishers maximize the price of their inventory while advertisers bid for individual impressions at prices that reflect the best value for them. This buying and selling mechanism happens in real time while a web-page of a publisher is being downloaded.

With the rise of search engines in the late 1900s, Sponsored Search has developed before Contextual Advertising [25]. Contextual Advertising, therefore, adopted many practices from Sponsored Search, such as characterizing textual ads using bidding phrases [3], which we observe in the current work. However, using bidding key-phrases without any context associated with them can lead to an interesting problem. Key-phrases here are search terms or words/phrases from the context of a page (such as title) or even the user profile.

1.1 The Problem

Contextual advertising helps by displaying ads that are relevant to the context at hand, thereby increasing the probability of an impression-conversion, while at the same time avoiding being too annoying or disinteresting to the user. However, on several occasions, the context can have a negative connotation, and displaying

Figure 1: Example of an unfortunate placement. [1]

advertisements that are relevant to this context might be detrimental to the advertiser. We refer to such a scenario as an *unfortunate placement* of an advertisement.

For example, in Fig. 1, the article describes the poor working conditions of Amazon warehouse workers in the UK. However, the placement of the advertisement for Amazon on this web page may negatively affect the advertiser. Furthermore, the user may be less compelled to click on the advertisement now, because of the negative correlation between the advertisement and the article.

We hypothesize that the root cause of these unfortunate placements is the lack of context associated with the triggering key-phrases. We believe that the sentiment carried by these phrases in the context of the web page will provide valuable information that could be used to mitigate such unfortunate placements. Therefore, we propose to use Sentiment Analysis to extract and analyze the sentiment of these triggering phrases and provide the missing context.

This work aims at applying Sentiment Analysis to intelligently filter out advertisements that may be harmed if they were matched to a given web page. In other words, given a web-page and a set of advertisements with their bidding key-phrases, our system selects relevant advertisements that may appear on the web page, without damaging their own brand image.

Modern online advertising architecture is complex and fast. It has a lot of underlying players who play different roles, from publishers who own the advertising inventory to advertisers who seek an audience for their products and services. The entire process of matching a user with a relevant advertisement usually takes place in less than 100 milliseconds [26]. Such strict time constraints make the current task at hand more challenging.

1.2 Problem Statement

In this work, we hypothesize that we could use the sentiment associated with a bidding phrase to avoid an unfortunate placement, in a reasonable time. Furthermore, we argue that such a system would also preserve existing good matches between positive pages and advertisements.

To demonstrate this, we propose the APNEA system, a light-weight system that extracts the sentiment from the triggering page P and associates it with candidate advertisers interested in advertising on P. By providing sentiment as the missing context to the advertisements, the system ensures that unfortunate placements are avoided. Being light-weight and simple, APNEA is fast and can be easily added on top of existing algorithms, without compromising the speed of the bidding process. Finally, advertisers are 'penalized' only by the sentiment context of the web page, therefore a non-negative context should not prevent a relevant advertisement from being displayed, ensuring that existing good matches are not disturbed in APNEA.

Experiments show that APNEA helps avoid unfortunate placements while maintaining ad-relevance. It outperforms several baselines in terms of accuracy on human-annotated test data while having a lower run-time, which is crucial for real-time bidding systems. Through this work, we also contribute two datasets to the scientific community for further research in this specific area: a dataset of advertisements, with bidding key-phrases and a manually annotated test dataset of page-ad matching ground-truth.

The rest of this work is organized as follows: In Section 2 we discuss the relevant background and related works. In Section 3, we give a detailed description of the APNEA system and the different sentiment analyzers used. In Section 4, we introduce the datasets used and evaluate the proposed system against a set of baselines and explore how different features of APNEA affect its performance on the test set. Finally, we conclude and shed some light on future work in Section 5.

2 BACKGROUND

The current work deals with the domains of online advertising and sentiment analysis. Therefore there are several works that are relevant to this task.

Langheinrich et al. [13] proposed ADWIZ, an unintrusive advertisement system based on a user's short term interests. However ADWIZ does not directly use the content of a web page. Ribeiro-Neto et al. [20] proposed ten strategies to solve the problem of page-ad matching, five of which are based on the cosine similarity between the page and ad vectors. Further more, to reduce mismatch

[1] Source: Reddit

between the vocabulary used in the web pages and the advertisements, the authors also suggest term expansion, with terms from pages that share common topics with the triggering page. Broder et al. [4] argue that "vagaries of phrase extraction and lack of context" may lead to irrelevant ads, which they propose to tackle by using both syntactic and semantic scores as part of the relevance score calculation. The syntactic score is calculated using cosine similarity between the page and ad vectors, while the semantic score is calculated based on the distance between two entities in a taxonomy. However, they do not consider the sentiment context of the triggering page in the page-ad matching process.

Qiu et al. [18] incorporate sentiment of the triggering page to provide the users better alternatives based on the aspect being criticized through their DASA system, by using rule-based keyword extraction to extract negative advertising keywords. These keywords are then used to place a rival brand that is better at the aspect towards which the page expresses a negative sentiment. DASA requires the advertisement data to be queryable by rival brands and aspects, making it difficult to incorporate into the modern advertising pipeline. Fan and Chang [7] propose SOCA, which removes negative sentences from the triggering page before page-ad matching using unigram SVM models to identify and extract sentiments. SOCA uses a linear combination of cosine similarity and the ontological similarity, not too different from [4]. Unlike DASA, SOCA does not need the advertiser data to be in a special format and therefore is practical to be deployed in the real world, making it the closest related work to our proposal and a strong baseline to compare against, boasting an accuracy of 68.2% on a test set of 150 blog pages.

Modern advertising architecture uses Real-time bidding for user behavioural targeting [25]. Besides work on page-ad relevance, Real-time bidding (RTB) has become a field of interest for various research groups such as in Information Retrieval to address the problem of ad relevancy [27], Data Mining to mine bidding patterns from a large stream of observed bids [6], and Machine Learning to learn models that optimize the campaign performance by bid estimation through click-through-rate (CTR) prediction [19]. Deman-Side Platforms (DSPs) process all the information regarding the user and the current URL and decide whether to bid or not for the current ad space through statistical analysis of past bids with the help of complex artificial intelligence systems [25]. More precisely, DSPs decide whether to bid on a particular ad inventory based on the predicted click-through-rate of the advertisement [21, 23, 25]. However, we do not have the data to build such a system. Several studies have shown that ad-relevance increases click-through-rate in advertisements [5, 24]. Therefore, we opt to use the relevance of an advertisement to the triggering page as a proxy for the click-through-rate thereby decide whether an advertiser would bid on the triggering page.

3 METHODOLOGY

In this section, we elaborate on the architecture behind our APNEA system and provide details regarding the Sentiment Analysis models that we incorporate into the proposed system. The APNEA pipeline consists of three significant stages, i.e. Advertisement Preprocessing, Sentiment Extraction and Page-Ad matching. Before we delve into the details of each stage, we would like to formally define the Problem Statement:

Given a web page P and a set of advertisements ADS, where each advertisement $adv_i \in ADS$ has a list of bidding phrases, $\{p_i^1, p_i^2, ...p_i^j\}$ with auxiliary information, select relevant advertisements $adv_{win} \subseteq ADS$ such that the sentiment associated with each advertisement $adv_k \in adv_{win}$ is non-negative, with respect to P.

3.1 Advertisement Pre-processing

The auxiliary information gives advertisers additional control on whether they care about the sentiment context of the bidding phrases, and the importance of the phrases to the advertisers' campaign. More precisely, each advertiser adv_i has a list of 3-tuples, (p_i^j, s_i^j, w_i^j), where p_i^j is the j^{th} bidding phrase, s_i^j is a Boolean flag, denoting whether the advertiser is sentiment-agnostic with respect to the current bidding phrase and w_i^j is a real number denoting the (commercial or semantic) importance of the bidding phrase to the advertiser. By default, all bidding phrases are sentiment-sensitive and carry a weight of 1.0.

The s flag in a bidding 3-tuple enables advertisers to disregard the sentiment and bid for a phrase regardless, e.g. phrases that are generally associated with a negative sentiment. On a similar note, we also incorporate a term-expansion mechanism, to reduce page-ad vocabulary mismatch. However, we believe that there may be significant noise in the expanded terms, which might disrupt the page-ad matching mechanism, therefore we discount the effect the expanded terms have on the final relevance score by reducing the weight associated with the expanded terms, by a reduction factor $r \in [1, \infty)$, a hyper-parameter of the system.

In order to determine the ad-relevance with respect to a given web page, we represent each advertiser as a unigram vector. We also use a mapping from a bidding keyword to an advertiser, to quickly identify advertisers who are bidding on a token.

The ad-vectors are constructed as follows: Each of the bidding phrases of an advertiser adv_i is pre-processed, with steps including lemmatization, converting to lowercase, and stop-word removal. The weights of the resulting tokens are then cumulatively added to the corresponding dimension in the vector space of the ad-vector, with a token that is derived from a bidding phrase p_i^j, contributing a weight of w_i^j. If term-expansion is enabled, the bidding phrase p_i^j is also queried on Wikipedia and if the corresponding article is not a disambiguation page, then the top 5 most frequent nouns are added as expanded terms to the ad-vector, each contributing a weight corresponding to w_i^j/r where r is the reduction factor.

A `keyword_to_advertiser` mapping is constructed by inserting the pre-processed keywords as keys into a map. The corresponding value for a given key in the map is a set of advertisers that are interested in that bidding keyword. If term expansion is enabled, the expanded terms that are extracted for the construction of the ad-vectors are also included in this dictionary.

Finally, for each advertiser, we keep track of the s-flag for the different bidding keywords. There are a few assumptions that we make at this step: Firstly, an advertiser is always sentiment-sensitive towards their own brand. Secondly, any expanded terms that are derived from a bidding phrase p_i^j will share the same s-flag, s_i^j. Thirdly,

if a bidding keyword or expanded keyword has two different *s*-flags in different 3-tuples, then the `true` flag will have priority over the `false` flag. This is to ensure that the advertiser's preference for sentiment-insensitivity prevails.

3.2 Sentiment Extraction from Triggering Page

In this section, we go into the details of the second component of the APNEA system – extracting the sentiments expressed in the triggering page.

The first step in this phase is to fetch the contents of the web page and clear unnecessary elements in the Document Object Model tree[14]. This can be achieved by any run-of-the-mill request package and an HTML parser. We define a *chunk* as a singular piece of text that is evaluated for sentiment, which is associated with the bidding keywords that appear in this chunk. We can tokenize the input document into chunks at different levels:

- **Document Level Tokenization:** In this approach, we treat the entire document as a single chunk and extract its sentiment.
- **Sentence Level Tokenization:** This approach treats each sentence of the document as a chunk, extracts its sentiment.
- **Sentence N-gram Tokenization:** In this approach, we treat each sentence n-gram in the document as a chunk. This is a heuristic approach to the problem of coreference resolution, under the assumption that the target nouns are local to each n-gram.

We hypothesize that the attention of the user varies across the content of the web page. Therefore, we assign a weight of *2.0* to the first chunk and equal weights of *1.0* to the rest of the web page. We wish to experiment with different distributions of these 'attention weights' in our future work.

In this work we employ many pre-processing steps like Text Tokenization, Lemmatization, Uniform Case conversion, Removing punctuation marks, Expanding tokens like *n't* to *'not'*, *'d* to *'would'* etc., and Remove stop-words. For lexicon-based approaches, we have an additional step where tokens that follow negative tokens such as *'not'*, *'never'*, *'no'*, etc. are combined to form a new token such as '*not_like*' from '*not like*'. These special tokens are handled internally by the Sentiment Analyzer and are assigned a sentiment vector that is the negation of the sentiment originally associated with the target token (*'like'*, in this example.) In this context, negation refers to interchanging the positive and negative scores in the 2-dimensional sentiment vector. If the target token is neutral, for the sake of simplicity, we revert back to the old technique of treating the two tokens separately.

Sentiments in this work are expressed as 2-dimensional sentiment vectors, where the values in each dimension correspond to *positive* and *negative* sentiment scores respectively. We opt to go for binary classification of the sentiment space, as *neutral* sentiments are also considered favorable for advertising. However, we ensure that they are differentiated from a positive sentiment, by assigning 0.5 points to the positive score of a neutral sentiment vector.

In this project, we chose to explore three different sentiment lexicons. We use a sentiment lexicon to determine the sentiment of a chunk of text by summing up the sentiment vectors of the constituent words. Text pre-processing steps discussed earlier are employed to avoid common pitfalls in text analysis. Opinion-Miner [11], SocialSent Lexicon from `r/news` [10], and SentiWordNet Lexicon [1]. We chose the `r/news` lexicon as the domain of our test set is news articles. The sentiment values are thresholded at zero and labelled as positive and negative. Also, APNEA considers only the positive and negative scores for the SentiWordNet Lexicon.

We also chose to include two ML-based approaches in our Sentiment Extraction subtask: an SVM model and the Stanford Sentiment Analyzer. Since we do not have any domain-specific training data to train our SVM model, we chose to use the Sentiment Labelled Sentence Dataset from UCI [12] for cross-domain training. The dataset consists of reviews from 3 major websites on the World Wide Web: Yelp, IMDB, and Amazon. We combine all the reviews and shuffle them randomly before training. We used the 50-dimensional GloVe [17] word embeddings and converted each sample to a sentence vector by averaging the constituent vectors. Our model reported a mean Accuracy of `76.4%` across 5 random train-test splits, with the best penalty parameter found to be `16.0`, using internal cross-validation. The Stanford Sentiment Analyzer [22] employs the Recursive Neural Tensor Network (RNTN) trained on the Stanford Sentiment Treebank [22]. The model achieves an accuracy of around 85% at sentence level on the Treebank test set. Besides its strong performance, the model is easily available through the Stanford CoreNLP Suite, making it easy to integrate into our experiments. The sentiment levels from the model are mapped to corresponding vectors: Very Positive ([2, 0]), Positive ([1, 0]), Neutral ([0, 0]), Negative ([0, 1]) and Very Negative ([0, 2]). For more details on the model architecture, we direct the reader to [22]. Note that APNEA can work with a lexicon-based as well as a machine learning-based sentiment extractor.

3.3 Page-Ad Matching

In this section, we describe in detail the last stage of the APNEA pipeline – page-ad matching. This stage comprises of computing the document vector from the sentiments extracted from the chunks and calculating the relevance scores between all advertisers and the document as the cosine similarity of the ad-vectors computed in Section 3.1 and the computed document vector. We also introduce two more functionalities of the APNEA system that further help it in solving the problem of unfortunate placements.

3.3.1 Blacklists. Understanding the sentiment associated with a web page is sometimes not enough to avoid an unfortunate placement. For example, a news article that talks about rising obesity rates in the United States may trigger an advertisement for fast food restaurants. In other cases, a brand may have a history of controversies on a particular topic that they would like to avoid. In such cases, a blacklist would come in handy to help an advertiser avoid an unfortunate placement in contexts that they do not wish to appear in. In our work, we use a list of keywords that an advertiser provides as a blacklist. During page-ad matching, whenever a blacklisted keyword has been encountered in the document, the advertisers that blacklisted the said keyword are marked and removed from the bidding process. Blacklist entries are optional for each advertiser. We examined a validation set (See Section 4) and added relevant blacklist entries to appropriate advertisers.

3.3.2 Targeted Sentiment. Targeted Sentiment identifies targets by parsing the title of the current article and matching them against its database of existing advertisers. If no targets are found, the system continues to evaluate all brands according to the sentiments reflected in the document. Otherwise, the relevance scores of all non-targeted advertisers are calculated on the absolute measure of the computed document vector. This has the effect of not penalizing the non-targeted advertisers based on the sentiment context of the document. The main assumption behind targeted sentiment is, given any targeted advertiser, we assume that its competitors have bid phrases similar to its own. Therefore, its competitors will also get triggered due to the semantics of the article. Since the competitors are not penalized, they get bumped up to a higher rank and have better chances of getting placed on the web page.

3.3.3 Page-Ad Matching Algorithm. The algorithm works as follows, the document retrieved is divided into *chunks*. For each chunk, we use the sentiment analyzer discussed in Section 3.2 to determine the sentiment of the chunk. The extracted sentiments are scaled according to the 'attention weights' described in the previous section.

The algorithm iterates over each token in a chunk and uses the mapping `keyword_to_advertiser`, to find the triggering keywords and the corresponding advertisers who are bidding on the said words. The bidding advertisers are added to a set to make up the candidate advertisers C. For each bidding keyword in the chunk, the extracted sentiment vector is added to the corresponding dimension of a document matrix M_d. It is also at this stage that the tokens are checked for any blacklisted key-phrases and the corresponding advertisers B are removed.

In order to compute the document vector from the matrix $M_d \in \mathbb{R}^{2\times|V|}$, where $|V|$ is the size of the vocabulary, we apply a scoring function column-wise on M_d. A scoring function takes a 2d sentiment vector and outputs a real value, reflective of the sentiment expressed by the input vector. If a candidate advertiser is sentiment-agnostic towards a set of keywords, the scores along those dimensions are taken as absolute values. Similarly, if targeted sentiment is enabled and the candidate advertiser is not a targeted advertiser, then the scores along all dimensions are taken as absolute values. Then, the relevance score for each candidate advertiser is calculated as the cosine similarity between the ad-vector constructed in Section 3.1 and the computed document vector. Finally, the advertisers are sorted in decreasing order of their relevance scores.

3.4 Scoring Functions

The system uses several different scoring functions which we adopt from [2]. More specifically, we use the Sentiment Difference (SD), Sentiment Maximum (SD), Threshold Difference (TD) and Threshold Maximum (TM) functions from [2]. We also propose a new scoring function, Logarithmic Ratio (RL) based on intuitive metrics and evaluate its performance against the rest.

Logarithmic Ratio (RL): The function evaluates a sentiment vector sv as follows, where p denotes the positive score and n denotes the negative score:

$$Score_{RL}(sv) = \begin{cases} max(p, \epsilon), & \text{if } n = 0 \\ p/n * \log(abs(p-n)+1) * sign(p-n), & \text{otherwise} \end{cases}$$

The function takes the maximum of the positive score and a small ϵ (default value 0.01) when the negative score is zero. Otherwise, it combines the ratio of positive to negative scores, with the difference function that grows sub-linearly and the polarity assigned by the sign of the difference function.

3.5 Real-time Performance

Real-Time Bidding (RTB) works in real-time and therefore, it is essential for any augmenting functionality to achieve the same run-time performance for practicality and compatibility. APNEA attempts to achieve this in several ways:

- Firstly, the system pre-processes the advertisements offline, therefore eliminating the need to expand terms and construct the ad-vector for each advertiser at run-time.
- Secondly, unlike SOCA [7], APNEA eliminates term expansion at the document level. This greatly reduces the processing time of each document, while not sacrificing much on accuracy since the ad-vectors are already term-expanded.
- Thirdly, the proposed system primarily utilizes lexicon-based sentiment analyzers, which are faster than ML-based approaches due to the absence of additional steps such as the conversion of input text to a feature vector. However, if a machine learning approach is fast enough, it can also be used. Deep learning approaches are known to be fast in their induction phase.
- Finally, APNEA is parallelizable. The document matrix construction can be mapped to multiple processes that handle different parts of the input document. The final matrix can be constructed by simple summation of the individual results. In a similar fashion, the relevance scores of each advertiser can also be computed parallelly, making APNEA practical to port to existing Demand-Side-Platforms (DSPs).

4 EXPERIMENTAL RESULTS

In this section, we go into the details of the evaluation of our system against baselines and explore how the different parameters affect the performance of the system on test data. We also discuss how the test set was collected and other experimental details. The code and data used for the evaluation is available at our online repository[2].

4.1 Data

We use a modified version of the Open Advertising Dataset [9] from Google, which consists of data related to Sponsored Search, with advertisers, landing page URLs, click-through-rates etc. More precisely, we use the web pages dataset, and edit the advertisers and the keywords, such that relevant articles could be found on Google News and the `r/news` subreddit. The final version of the advertisements dataset consisted of 68 advertisers, each advertising a single advertisement. Note that since we edit the original dataset, the supporting information such as the click-through-rates are no longer applicable to our dataset.

Since APNEA does not have any learning component, we do not have any training data in our experimental setup. However, we use a small additional dataset of 25 news articles as a validation

[2] https://github.com/blumonkey/apnea

set to explore and tweak the different parameters of the system. The validation set was used to develop the strategy of targeted advertising, blacklist and reduction factor.

The test set consists of 177 articles that were selected manually using Google News Search and Reddit `r/news` Search. The dataset was annotated by 3 annotators based on the criterion: 'Is this a good place to advertise my brand?'. The annotators worked independently and the results are consolidated by taking the majority vote. The average pair-wise Kappa agreement coefficient is 0.87. The details of the validation set and the test set are found in Table 1:

Data Set	Articles	Sentences	Tokens
Validation	25	685	13940
Test	177	4381	93119

Table 1: Data Set Statistics

Each instance in either set consists of a URL which corresponds to the triggering document, and a list of advertisers, each of which is marked with one of three different labels: `UNSAFE`, `SAFE` and `DONT_CARE`. Advertisers marked `UNSAFE` are those that the annotators believe will get hurt if their brand is placed on the triggering document. Advertisers marked `SAFE` are those that are not only safe to be placed on the triggering document but are also relevant to the context of the document. Advertisers marked `DONT_CARE` are those who are not relevant to the context of the triggering document but may not be harmed when advertised there. A ranking is desired if none of the `UNSAFE` advertisers are in the top-5 while at least one of the `SAFE` advertisers is in the top-5. Note that there may be samples where no advertiser is `UNSAFE` (or `SAFE`).

This leads to two different types of errors that we come across in our evaluation.

- **Type 1 Errors** ($type_1$): These are the samples where at least one `UNSAFE` advertiser is in the top-5 ranks. In other words, this is a false-positive ranking where an `UNSAFE` advertiser is marked safe to advertise.
- **Type 2 Errors** ($type_2$): These are the samples where none of the `SAFE` advertisers are in the top-5 ranks. In other words, this is a false-negative ranking where a `SAFE` advertiser is falsely marked as `UNSAFE` or `DONT_CARE`.

We use accuracy as the metric of choice because the outcome of the proposed system is not the individual labels of the advertisements, but rather the ranking as a whole. The accuracy of a system can be calculated as follows:

$$Accuracy = \frac{total - type_1 - type_2}{total}$$

4.2 Experimental Results

4.2.1 Baselines. We compare our system against two baselines, namely the traditional Contextual Advertising System (CA) and SOCA by Fan and Chang [7]. CA is implemented using APNEA without the sentiment component. The system computes the term-frequency (tf) vector of the document as the document vector and the cosine similarity between the document and the ad-vectors is used to calculate relevance. We also include two baselines, BL_POS and BL_NEG, where all sentiments are treated as positive and negative, respectively. The reasoning behind taking BL_POS / BL_NEG as the baseline algorithms is that BL_POS acts like a traditional contextual advertising system where the sentiment of the document is ignored. BL_NEG acts as a risk-averse alternative, where the advertisers may lose placement, even if the document is relevant and the sentiment is positive. All baselines based on APNEA have term expansion and *s*-flag enabled, using the Threshold Difference (TD) scoring function and working at Sentence level analysis, with a reduction factor of 1.0. Enabling the *s*-flag functionality respects the individual sentiment-sensitivity settings of the advertisers.

SOCA is implemented as described in [7], with the best parameters and default values described in the work. We constructed the *tf-idf* vectors for both the advertisements and the test documents to calculate the cosine similarity component of the scoring metric used by SOCA. However, they perform poorly at the current task, therefore, we also use 300-dimensional word2vec embeddings [16] and average the embeddings of all tokens in a document to construct its representation. Furthermore, the authors of SOCA [7] do not define any default value for parameter δ in the web-based term expansion stage of the SOCA pipeline. Since the possible values for δ are in the range of $(0, \infty)$, we experimented with a few values and observed that they add noise to the system's performance. Therefore, we eliminate the web-based term expansion of SOCA. Furthermore, SOCA reports its best accuracy at 68.2% using an SVM based sentiment analyzer that was trained on reviews from *epinion.com*. Since the data is no longer available at the website, we use our SVM model from Section 3.2, trained on the UCI dataset as the sentiment analyzer for SOCA.

In order to make the comparison between our system and the baselines fair, we use the same configuration as the BL_POS / BL_NEG baseline, with the only difference being that sentiment is now extracted using the Opinion Mining Sentiment Lexicon. We represent this configuration of APNEA as Basic Conf. in Table 2. In order to remove the effect of the sentiment lexicon used, we also run SOCA on the Opinion Mining Lexicon [11], denoted as SOCA w/ OM.

Table 2 shows the performance of the different baselines on the test set and how APNEA out-performs all baselines with a significant margin. Our APNEA system, compared against the above two, shows that taking the sentiment into context, we not only avoid unfortunate placements (as compated to BL_POS), we also maintain ad-relevance (as compared to BL_NEG). SOCA with *tf-idf* vectors and web-expansion performs at par with the regular CA system. We suspect this is because of the noise introduced by web-expansion and the *tf-idf* vectors not capturing the document/ad representation to the fullest extent on a small dataset. The version of SOCA with Opinion Mining Lexicon is not more effective than APNEA, which suggests that the difference in the methodology is the primary player in these results.

4.2.2 Variations in APNEA. We run experiments with increasing functionality of the system to demonstrate the effect of each feature. We also explore the effect of different sentiment analyzers, scoring functions, and analysis levels on the APNEA system.

Table 3 shows the effect of Targeted Sentiment and Blacklist on the performance of the APNEA system. We attribute the small

Models	$type_1$ Errors	$type_2$ Errors	*Accuracy*
CA	73	8	0.54
BL_POS	75	**6**	0.54
BL_NEG	**0**	104	0.41
SOCA w/ SVM, *tf-idf* Web-expansion Enabled	71	9	0.55
SOCA w/ OM, *tf-idf* Web-expansion Enabled	71	9	0.55
SOCA w/ SVM, word2vec Web-expansion Disabled	26	57	0.53
SOCA w/ OM, word2vec Web-expansion Disabled	23	62	0.52
APNEA (Basic Conf.)	9	27	**0.80**

Table 2: Errors and Accuracy for the Baseline Models

Models	$type_1$ Errors	$type_2$ Errors	*Accuracy*
At Sentence Level Analysis			
APNEA (Basic Conf.)	**9**	27	0.80
APNEA (w/ TS)	11	26	0.79
APNEA (w/ TS & BLST)	11	26	0.79
APNEA (w/ TS & BLST, $r = 2.0$)	12	**23**	**0.80**
At Document Level Analysis, $r = 2.0$			
APNEA (DOC.)	5	30	0.80
APNEA (DOC w/ TS)	7	26	0.81
APNEA (DOC w/ TS & BLST)	7	26	0.81

Table 3: Errors and Accuracy with Targeted Sentiment and Blacklist

decrease in the accuracy at Sentence Level Analysis with Targeted Sentiment, to the samples in the dataset where target extraction from the title was inaccurate. The blacklist constructed from the small validation set proves to be ineffective on the larger test set. However, the effect of the reduction factor is evident from the reduced number of $type_2$ errors, resulting in improved accuracy.

In the case of Document Level Analysis, we can see that Targeted Sentiment, boosts the accuracy of the system, albeit by a small percentage. This is because, in document level analysis, all the triggering keywords get associated with the same sentiment i.e. the sentiment of the document. By using Targeted Sentiment, the sentiment of the document reflects only on the advertisers that are targeted in the title, letting competitors to secure a position in the top-5 ranks by relevance. The effect of the blacklist on the performance of the system at this level is again non-existent.

Models	$type_1$ Errors	$type_2$ Errors	*Accuracy*
Variations in Sentiment Analyzers			
APNEA (SentiWordNet)	29	**19**	**0.73**
APNEA (SSA)	**2**	91	0.47
APNEA (SocialSent)	40	31	0.60
APNEA (SVM)	5	50	0.69
Variations in Scoring Functions			
APNEA (TM)	**12**	**22**	**0.81**
APNEA (RL)	20	25	0.75
APNEA (SD)	12	27	0.78
Variations in Analysis Level			
APNEA (SENT_NGRAM, $n = 3$)	6	**29**	**0.80**
APNEA (SENT_NGRAM, $n = 5$)	**5**	32	0.79

Table 4: Errors and Accuracy at Other Configurations

We also explore other configurations where the Sentiment Analyzer, Analysis Level and Scoring function have been changed. The base model for these variations is the APNEA with Opinion Mining Lexicon at Sentence Level Analysis, using the TD scoring function. All the results presented in Table 4 are at a reduction factor $r = 2.0$ with *s*-flag, Targeted Sentiment and Blacklist enabled.

4.3 Run-time Analysis

In order to demonstrate that our APNEA is faster than the SOCA framework, we evaluate the run-times of both the systems on a Quad-Core 7th Gen. i5 Processor@2.5 GHz with 8GB memory. APNEA system uses the Opinion Mining Lexicon at Sentence and Document Level Analysis, with the *s*-flag, Targeted Sentiment and Blacklist enabled, with a reduction factor $r = 2.0$. To keep the comparison fair, the SOCA system is also evaluated on the Opinion Mining Lexicon, without web expansion enabled on the input document. Both systems are evaluated on 25 random URLs from the test set, across 5 independent runs.

In order to reduce the implementation bias in SOCA, both systems have pre-processing of the advertisements done offline, and the time taken for fetching the Wikipedia articles on the document expansion stage for SOCA has been excluded from the run-time computation. However, we did not index the document synsets as suggested in [7], because documents on the World Wide Web

Model Name	Time (*s*)					Mean
APNEA (SENT.)	6.05	5.94	5.92	5.91	5.89	5.94
APNEA (DOC.)	2.68	2.64	2.55	2.60	2.54	**2.60**
SOCA	36.09	40.25	46.72	40.41	56.16	43.93

Table 5: Run-times across the runs, using SOCA and APNEA

are highly volatile and dynamic and indexing information on the document side does not seem practical in a real-world scenario.

Table 5 shows the run-times of each system. It is evident that our proposed system is faster than SOCA, even without web expansion enabled in the latter.

5 CONCLUSIONS AND FUTURE WORK

In this work, we defined the problem of *unfortunate placement* and tried to address it using the APNEA system we propose. We first introduced the problem statement and the challenges associated with it, namely avoiding an unfortunate placement and at the same time, maintaining ad-relevance with respect to other advertisers. We then discussed how Sentiment Analysis can help provide the right context to avoid an unfortunate placement. We go into the details of the proposed system, and evaluate it against set baselines, on a human-annotated dataset. Experimental results show that our proposed system outperforms the defined baselines by a significant margin on the annotated dataset. Run-time analysis shows that our system is also faster than the baseline SOCA system by up to 94%.

The present work has many challenges, in the fields of Real-time bidding and Sentiment Analysis. We use Sentiment Analysis to provide the necessary context to APNEA, but we would like to extend this to Emotion Mining, which provides much more granularity and control to the advertisers. For this, we could use a lexicon-based or a machine learning-based emotion mining approach [8]. Due to the lack of annotated data, we have primarily preferred a lexicon-based approach to the Sentiment Analysis component of APNEA. With the availability of domain-specific annotated training data, we would like to test how a learning-based approach performs against the lexicon-based approach.

REFERENCES

[1] Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani. 2010. SentiWordNet 3.0: An Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining. In *Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC'10)*. European Languages Resources Association (ELRA), Valletta, Malta, 2200–2204.

[2] AR Balamurali, Subhabrata Mukherjee, Akshat Malu, and Pushpak Bhattacharyya. 2012. Leveraging sentiment to compute word similarity. In *6th International Global Wordnet Conference (GWC)*. Citeseer, Matue, Japan.

[3] Andrei Broder, Marcus Fontoura, Vanja Josifovski, and Lance Riedel. 2007. A Semantic Approach to Contextual Advertising. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '07)*. ACM, New York, NY, USA, 559–566. https://doi.org/10.1145/1277741.1277837

[4] Andrei Broder, Marcus Fontoura, Vanja Josifovski, and Lance Riedel. 2007. A semantic approach to contextual advertising. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, Amsterdam, The Netherlands, 559–566.

[5] Patrali Chatterjee, Donna L Hoffman, and Thomas P Novak. 2003. Modeling the clickstream: Implications for web-based advertising efforts. *Marketing Science* 22, 4 (2003), 520–541.

[6] Ying Cui, Ruofei Zhang, Wei Li, and Jianchang Mao. 2011. Bid landscape forecasting in online ad exchange marketplace. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, San Diego, USA, 265–273.

[7] Teng-Kai Fan and Chia-Hui Chang. 2010. Sentiment-oriented contextual advertising. *Knowledge and information systems* 23, 3 (2010), 321–344.

[8] Ameneh Gholipour Shahraki and Osmar R. Zaiane. 2017. Lexical and Learning-based Emotion Mining from Text. In *18th International Conference on Intelligent Text Processing and Computational Linguistics*. Budapest, Hungary.

[9] Google. [n.d.]. *Open advertising dataset - Google Code Archive*. Google Code.

[10] William L Hamilton, Kevin Clark, Jure Leskovec, and Dan Jurafsky. 2016. Inducing domain-specific sentiment lexicons from unlabeled corpora. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing. Conference on Empirical Methods in Natural Language Processing*, Vol. 2016. NIH Public Access, 595.

[11] Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 168–177.

[12] Dimitrios Kotzias, Misha Denil, Nando De Freitas, and Padhraic Smyth. 2015. From group to individual labels using deep features. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 597–606.

[13] Marc Langheinrich, Atsuyoshi Nakamura, Naoki Abe, Tomonari Kamba, and Yoshiyuki Koseki. 1999. Unintrusive customization techniques for web advertising. *Computer Networks* 31, 11-16 (1999), 1259–1272.

[14] Philippe Le Hégaret. 2002. *The W3C Document Object Model (DOM), World Wide Web Consortium*. https://www.w3.org/2002/07/26-dom-article.html

[15] Google LLC. 2018. *Alphabet Announces First Quarter 2018 Results*. https://abc.xyz/investor/pdf/2018Q1_alphabet_earnings_release.pdf

[16] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. *CoRR* abs/1301.3781 (2013). arXiv:1301.3781 http://arxiv.org/abs/1301.3781

[17] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global Vectors for Word Representation. In *Empirical Methods in Natural Language Processing (EMNLP)*. 1532–1543. http://www.aclweb.org/anthology/D14-1162

[18] Guang Qiu, Xiaofei He, Feng Zhang, Yuan Shi, Jiajun Bu, and Chun Chen. 2010. DASA: dissatisfaction-oriented advertising based on sentiment analysis. *Expert Systems with Applications* 37, 9 (2010), 6182–6191.

[19] Kan Ren, Weinan Zhang, Yifei Rong, Haifeng Zhang, Yong Yu, and Jun Wang. 2016. User response learning for directly optimizing campaign performance in display advertising. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*. ACM, 679–688.

[20] Berthier Ribeiro-Neto, Marco Cristo, Paulo B Golgher, and Edleno Silva de Moura. 2005. Impedance coupling in content-targeted advertising. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 496–503.

[21] Matthew Richardson, Ewa Dominowska, and Robert Ragno. 2007. Predicting clicks: estimating the click-through rate for new ads. In *Proceedings of the 16th international conference on World Wide Web*. ACM, 521–530.

[22] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*. 1631–1642.

[23] Ilya Trofimov, Anna Kornetova, and Valery Topinskiy. 2012. Using boosted trees for click-through rate prediction for sponsored search. In *Proceedings of the Sixth International Workshop on Data Mining for Online Advertising and Internet Economy*. ACM, 2.

[24] Chingning Wang, Ping Zhang, Risook Choi, and Michael D'Eredita. 2002. Understanding consumers attitude toward advertising. In *Americas Conference on Information Systems*. AIS Electronic Library, 1143–1148.

[25] Jun Wang, Weinan Zhang, and Shuai Yuan. 2016. Display advertising with real-time bidding (RTB) and behavioural targeting. *arXiv preprint arXiv:1610.03013* (2016).

[26] Jun Wang, Weinan Zhang, and Shuai Yuan. 2016. Display Advertising with Real-Time Bidding (RTB) and Behavioural Targeting. *CoRR* abs/1610.03013 (2016). arXiv:1610.03013 http://arxiv.org/abs/1610.03013

[27] Weinan Zhang, Lingxi Chen, and Jun Wang. 2016. Implicit Look-alike Modelling in Display Ads: Transfer Collaborative Filtering to CTR Estimation. *CoRR* abs/1601.02377 (2016). arXiv:1601.02377 http://arxiv.org/abs/1601.02377