Natural Language Annotations for Search Engine Optimization

Porter Jenkins[†], Jennifer Zhao[‡], Heath Vinicombe[‡], Anant Subramanian[‡], Arun Prasad[‡], Attila Dobi[‡], Eileen Li[‡], Yunsong Guo
[†]Pennsylvania State University, [‡]Pinterest
[†] prj3@psu.edu
[‡]{jzhao, heathvinicombe, asubramanian, aprasad, adobi, eileenli, yunsong}@pinterest.com

ABSTRACT

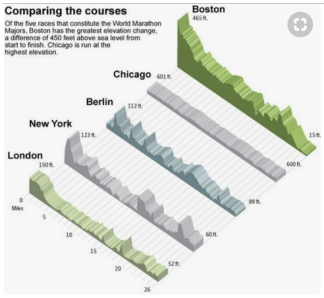
Understanding content at scale is a difficult but important problem for many platforms. Many previous studies focus on content understanding to optimize engagement with existing users. However, little work studies how to leverage better content understanding to attract new users. In this work, we build a framework for generating natural language content annotations and show how they can be used for search engine optimization. The proposed framework relies on an XGBoost model that labels "pins" with high probability phrases, and a logistic regression layer that learns to rank aggregated annotations for groups of content. The pipeline identifies keywords that are descriptive and contextually meaningful. We perform a large-scale production experiment deployed on the Pinterest platform and show that natural language annotations cause a 1-2% increase in traffic from leading search engines. This increase is statistically significant. Finally, we explore and interpret the characteristics of our annotations framework.

ACM Reference Format:
Porter Jenkins[†], Jennifer Zhao[‡], Heath Vinicombe[‡], Anant Subramanian[‡], Arun Prasad[‡], Attila Dobi[‡], Eileen Li[‡], Yunsong Guo . 2020. Natural Language Annotations for Search Engine Optimization. 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/10.1145/3366423.3380049

1 INTRODUCTION

Understanding content at scale is a difficult yet important problem for many organizations. A system that can automatically and effectively understand content is better at connecting users with what they care about. In turn, this drives retention and new user growth. One approach to scalable content understanding is through the extraction of natural language key phrases. These key phrases characterize and describe the content in a way that a human can easily understand. In this paper, we study how natural language annotations can help drive new user traffic via search engine optimization (SEO).

The virtue of relying on a natural language approach to characterize content, is that the output can easily be represented in a way that is both human and machine readable. A vector of natural language key phrases and their corresponding scores can be treated as a distribution of topics [3], or as an embedding vector [6]. An algorithm can then compute item-to-item similarities using a vector similarity measure. Moreover, a human can easily understand the output in his or her own natural language. Such annotations can better capture the interest of new and existing users by summarizing key features of content, which drives user engagement and growth.



Annotation	Score
marathon	0.97
boston marathon	0.95
marathon running	0.93
nyc marathon	0.91
chicago marathon	0.90
boston marathon inspiration	0.90
marathon training	0.89
half marathon	0.87
marathon prep	0.87
half marathon training	0.85
marathon preparation	0.84
marathon tips	0.84
berlin marathon	0.84
boston	0.83
running	0.81
running workouts	0.81
first marathon	0.80
marathon inspiration	0.80
trail running	0.77
run runner	0.75
sprinting	0.73
running motivation	0.72
runner	0.70
running tips	0.69
jogging	0.65

Figure 1: An example "pin" and its corresponding top 25 annotations. This pin contains an image comparing five popular marathon courses (left). We also display the pin annotations that best describe this content (right). Each annotation, is a short, n-gram phrase with a corresponding confidence score. The top-ranked annotations include, "marathon", "boston marathon", etc.

Most existing work focuses on how to better engage users within a platform. For instance, classical matrix factorization methods embed users and items to recommend relevant content to users [10]. More recently, users and items are embedded using deep neural networks [11] for recommendation. Less work has studied how to best leverage content understanding to attract new users. Organic traffic from search engines is a key mechanism through which platforms can attract new users. The process of increasing organic search engine traffic is known as search engine optimization (SEO). One key strategy to SEO is to produce content keywords, or meta tags, that match popular search queries [1]. Meta tags are short phrases embedded in source code that describe a page to the search engine [14]. Therefore, in this paper we study the problem of automatically generating content annotations that can serve as meta tags for search engine optimization.

However, understanding content with natural language in a way that drives traffic from search engines is challenging. For one, it is difficult to identify what features of keyword phrases will cause a potential user to click a link to the platform. Such a decision process is subjective and user specific. Thus, in many cases there is no unambiguously correct answer; one of many keyword phrases could equally describe a picture, text, or video. Additionally, generating annotations at scale, and across many language, is difficult.

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

ACM ISBN 978-1-4503-7023-3/20/04.
https://doi.org/10.1145/3366423.3380049

Therefore, we design a framework to solve these challenges. We design a pipeline that learns to label content with semantically meaningful and descriptive phrases. By relying on an XGBoost model that labels content with text phrases, an aggregation layer, and a logistic regression layer that learns to rank, our pipeline produces content annotations that are descriptive and leverage user-defined context. We deploy an A/B test of our pipeline at Pinterest, one of the world's largest content recommendations systems and home to over 300 million users.

In summary, our contributions are:

- We propose a framework for generating keyword phrases for content called 'annotations'
- The proposed pipeline identifies semantically meaningful phrases that are 1) descriptive and specific, and 2) leverage important contextual information
- We deploy an extensive, online experiment on the Pinterest platform and show that, when used as meta tags for SEO, the proposed annotation pipeline increases traffic from Google search 1-2%, which is a statistically significant result

The rest of the paper is organized as follows: section 2 formalizes and defines the problem, section 3 provides a detailed description of the proposed framework, section 4 presents experimental results, section 5 discusses related work, and section 6 summarizes our findings.

2 PROBLEM DEFINITION

The primary goal of the current work is to produce natural language keywords that describe a set of user generated content. These "annotations" should be both accurate and specific. In this section we define the content annotation problem in terms of inputs and outputs. The input to our problem is a set of pins, and set of boards. The output is a ranked list of annotations for each board.

Definition 2.1 (Pin). A "pin" is a bookmarked webpage that collects important metadata from the page. We observe a set of pins, $\mathcal{P} = \{p_i : i <= n\}$, where n is the total number of pins. Each p_i is composed of important metadata including an image, link, title, and description, $p_i = \langle \mathbf{d}_i, b_j \rangle$ where $\mathbf{d}_i$ is a vector of metadata and b_j is an associated board.

Definition 2.2 (Board). A "board", b_j is a collection of content that a user finds interesting. The user might save content related to an idea he or she wishes to create in real life, or simply as a digital scrapbook. In our problem, we have a set of $\mathcal{B} = \{b_j : j <= m\}$, where each board, b_j is associated with a set of pins, $b_j = \{p_1, p_2, ...p_n\}$. We also observe board metadata such as title, or description.

Definition 2.3 (Annotations). An "annotation", a_k is a natural language keyphrase, typically one to six words in length, that describes a pin or a board. For each content type (e.g., board, pin), we observe a set of q relevant candidate annotations, $\mathcal{A} = \{a_k : i <= q\}$. We denote board annotations as $\mathcal{A}_{b_j} = \{a_1, ...a_q\}$ and pin annotations, $\mathcal{A}_{p_i} = \{a_1, ...a_q\}$.

Definition 2.4 (Content Annotation for SEO). The content annotation problem is defined as specifying a function, f, that maps a set of annotations to an ordered list of length, t. The list, $\mathcal{A}_{p_i}$, contains the top t most relevant annotations to board, p_i.

$$\mathcal{A}^*_{p_i} = f_p : A_{p_i} \rightarrow [a^{(1)}, a^{(2)}, ..., a^{(t)}] \quad (1)$$

Similarly, the list, $\mathcal{A}_{b_j}$, contains the top t most relevant annotations to board, b_j.

$$\mathcal{A}^*_{b_j} = f_b : A_{b_j} \rightarrow [a^{(1)}, a^{(2)}, ..., a^{(t)}] \quad (2)$$

The ultimate goal of the content annotation problem, is to produce a list $\mathcal{A}^*_{b_j}$ that is descriptive and captures semantic meaning of board, b_j.

3 METHODS

In this section, we propose a framework to solve the content annotation problem. Our pipeline consists of two stages. First, we extract pin-level annotations for fine-grained content understanding. Second, we use two aggregation layers to produce annotations that are more broadly relevant to the entire set of board-level annotations. An overview of the entire pipeline is presented in figure 2.

3.1 Pin Annotations

We first present our framework for extracting pin-level annotations, $\mathcal{A}_{p_i}$ [13]. We discuss the annotations dictionary, candidate generation, and the XGBoost model used for annotation scoring.

3.1.1 Dictionary. Annotations are limited to a finite vocabulary which we refer to as the annotations dictionary. The advantage of using such a dictionary over allowing annotations to be arbitrary ngrams is that it guarantees the annotations will be valid and useful phrases instead of misspellings (e.g., "recipies"), stopwords (e.g., "the"), fragments (e.g., "of liberty") and generic phrases (e.g., "ideas", "things"). The dictionary contains popular topics that were manually entered by users as search queries, hashtags, etc. A significant amount of human curation has gone into building the dictionary to ensure its quality is maintained, and we periodically use heuristics to trim out bad terms and use a spell checker to remove misspellings. We have around 100,000 terms in the dictionary for each language.

3.1.2 Candidate Generation. From the dictionary described in section 3.1.1, we generate a set of potential annotation candidates for each pin, $\mathcal{A}_{p_i}$. First, a text language detector determines the language of the text. Next each text string is tokenized with a language-specific tokenizer. A sliding window is then used to generate all ngrams containing between 1 and 6 words in the text. The ngrams are normalized by stripping out accents and punctuation and then stemming or lemmatizing depending on the language. The extracted annotations are canonicalized to reduce duplication (e.g., "sloth" is canonicalized to "sloths" to reduce redundant annotations). We maintain these canonical mappings in the dictionary.

3.1.3 Model. We use an XGBoost classifier to produce relevance scores for each annotation candidate $a_k \in \mathcal{A}_{p_i}$. The model is trained using a crowd-sourced data set where human annotators are asked to judge for a given pin, annotation pair, $\langle p_i, a_k \rangle$ whether the annotation is relevant to the pin; e.g., $y \in \{0, 1\}$. Around 150,000 labels per language comprise our training data set.

Once a set of candidate annotations, $\mathcal{A}_{p_i}$, has been generated we extract annotation-level features and feed them into an XGBoost

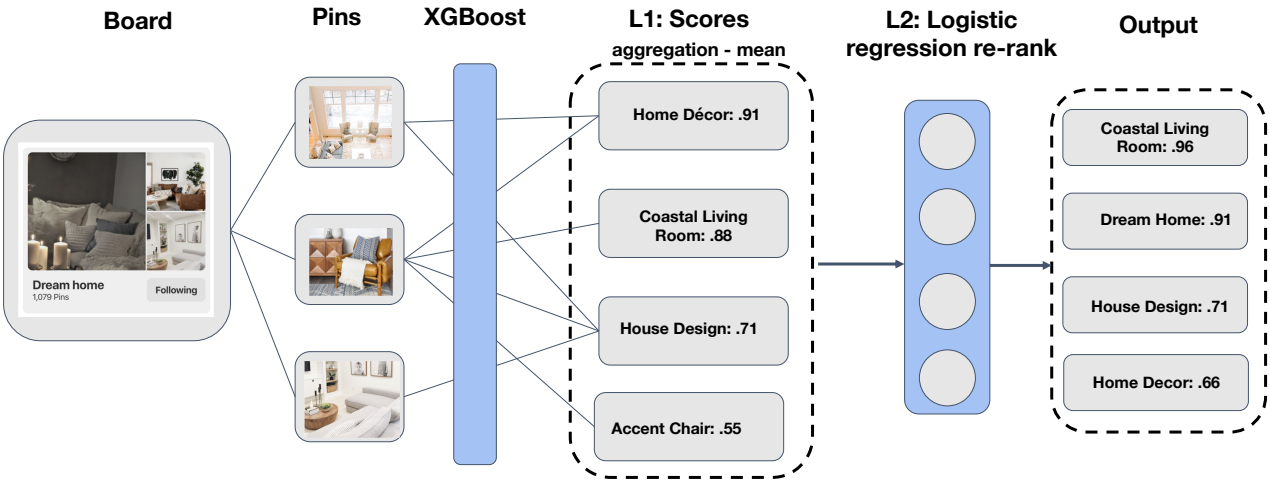


Figure 2: The proposed content annotation pipeline. A given board is associated with a collection of pins. First, for each pin we generate candidate annotations from a dictionary. We then extract features for each candidate pin-annotation pair (e.g., TF-IDF, cosine similarity). We then input the featurized pairs to an XGBoost classifier trained to predict whether the phrase is relevant or not. To summarize this fine-grained content understanding, we average these scores over pins to the board level. We then train a logistic regression layer to learn to rank the annotation scores such that descriptive and contextual annotations are surfaced.

classifier for scoring. The features we use in our XGBoost model are the following:

- **TF-IDF** Term Frequency - Inverse Document Frequency
- **Embedding similarity** — cosine similarity between a pin embedding and annotation embedding. The annotation embedding is a pretrained ConceptNet embedding [12] and the pin embedding is the average of the annotation embeddings for the given pin's candidate annotations

Once the features, $\mathbf{x}_k$, for a candidate annotation have been extracted, the samples are fed into the XGBoost classifier to produce a predicted probability, or score, $\hat{s}_{ik}$, that the annotation, a_k, is relevant to pin p_i.

$$\hat{s}_{ik} = P(y_{ij} = 1|\mathbf{x}_k) \tag{3}$$

The higher the value of $\hat{s}_{ik}$, the better the annotation is at describing the content of the corresponding pin. We can sort annotations on $\hat{y}_{ik}$ and take the top k to produce the more relevant pin annotations, $\mathcal{A}^*_{p_i}$.

3.2 Board Annotations

Previously, we discussed computing pin-level annotations, $\mathcal{A}^*_{p_i}$. These annotations are very fine-grained and describe individual pieces of content. Next, we complete our pipeline by aggregating fine-grained pin annotations to the board level. Doing so allows us to summarize many co-occuring items of content simultaneously. This framework consists of two key layers. In layer one, we perform a naive aggregation wherein we compute an annotation score averaged over pins. Next, we train a learning-to-rank logistic regression model to sort the annotation scores and surface more descriptive and more relevant annotations.

3.2.1 Naive Aggregation. To aggregate pin annotations to the board level we compute the arithmetic mean over pins:

$$s^{(1)}_{jk} = \frac{1}{n}\sum_{i=1}^{n} s_{ijk} \tag{4}$$

Where, $s^{(1)}_{jk}$ denotes the score. This simple aggregation summarized the pin-specific information to the board-level. For simplicity, we refer to this as the *layer-1 score*, or L-1 score. The superscript in equation 4 refers to the first ranking layer. However, aggregating by the mean scores alone tends to yield very broad, or generic annotations. For example, the top annotation in figure 1 is "marathon", a somewhat broad term. Perhaps a more descriptive phrase such as "marathon courses" would be a better description of the pin. We need a way of intelligently ranking the board annotations without losing important term specificity.

3.2.2 Learning-to-rank model. To this end, we train a learning-to-rank logistic regression model that predicts whether or not a given annotation, a_{jk} is relevant to the entire board. Similar to section 3.1.3, we crowdsource a human labelled data set where annotators are asked to label pairs of annotations. The task for the labeller is then to choose which of two annotations better describes the board content. Given two annotations, a_{j1} and a_{j2} for board b_j, the relevance determines the label. If $\text{rel}(a_{j1}, b_j) > \text{rel}(a_{j2}, b_j)$, then $y_{jk} = 1$ and 0 otherwise. This pairwise approach is useful because people tend to be better at judging the relative relevance of items, rather the absolute relevance [10] [5].

With this pairwise data set we can train our ranking model. We specify the model to learn the relevance:

$$s^{(2)}_{jk} = P(y_{jk} = 1|\mathbf{x}_{j1}, \mathbf{x}_{j2}) \tag{5}$$

We difference the feature values to compute the pairwise ranking scores following the **RankNet** procedure [4]:

$$s_{jk}^{(2)} = \sigma(g(\mathbf{x}_{j1}) - g(\mathbf{x}_{j2})) \quad (6)$$

where g is a linear scoring function $g(\mathbf{x}) = \mathbf{x}_{jk}\mathbf{w}^\top + b$. We refer to this quantity, $s_{jk}^{(2)}$, as the L-2 score, or the score from the second ranking layer. The main purpose of the model in equation 5 is to predict annotation relevance conditioning on the mean score computed in 4, as well as other features that might help the model discover more specificity. We explore other features in our experiments section (4.1).

3.2.3 Output. The proposed framework described above outputs a list t of annotations for each board, ranked by relevance score: $\mathcal{A}_{b_j}^* = [a^{(1)}, a^{(2)}, ..., a^{(t)}]$. This computed relevance score captures term specificity as well as important contextual information.

4 EXPERIMENTS

In this section, we first discuss our training and selection procedure for the ranking model described in equation 5. We then deploy the proposed pipeline into the Pinterest ecosystem for 30 days and observe the differential impact on search engine traffic. Finally, we provide a case study of a board and its corresponding annotations.

4.1 Ranking Model Selection

The learning-to-rank model in equation 5 models the board-level annotation relevance score, $\hat{y}_{ik}$, as a function of annotation features. Using the data set described in section 3.2.2 we perform an offline experiment to choose the optimal feature set. Using 5-fold cross-validation, we test five feature sets and a naive baseline in terms of accuracy, ROC (AUC), precision, and recall.

4.1.1 Features. We evaluate combinations of the following features:

- **L1-score** The L-1 scores computed over pins in equation 4.
- **Log IDF** The log of the inverse document frequency computed over the dictionary. Words that are more rare, or possibly more specific, have a higher IDF.
- **Term in board** Whether or not the annotation, a_{jk} appears in the board title or board description. This feature captures important contextual clues. In many cases, the user provides text that is a useful description of the content.
- **Term in board * log IDF** An interaction between 'term in board' and log IDF. Intuitively, if a word is rare and also appears in the board title or description, it will receive extra weight.
- **L1-score * log IDF** An interaction between 'term in board' and the L1-score. If a word has a high L-1 score and is also rare, it should receive extra weight.

In addition to different combinations of these features, we also compare to a naive baseline, which is to always choose the annotation with the higher L-1 score.

4.1.2 Results. The results are reported in table 1. Both the mean and standard deviation across each fold of the cross-validation scheme are provided. In general, the naive baseline that does not use the logistic ranking model performs poorly at predicting the labels from human annotators. It's prediction accuracy and ROC (AUC) are both around 0.5. The logistic regression ranking performs fairly well across all feature sets. In particular, the combination of L-1 score, log IDF, 'term in in board', and the interaction between L-1 score and log IDF performs better than or equal to all other models in terms of accuracy, ROC (AUC), and precision. Specifically, we observe that the addition of the interaction term gives it a slight improvement over the more simple model in the second row of table 1. This suggests that phrases that are both rare and have high average score from the first layer are predictive of what human annotators think. This helps the learning-to-rank model discover key terms that are also specific. We choose this model for deployment in a our online experiment.

4.2 Online SEO experiment

We deploy the annotations in a large-scale production environment at Pinterest for a period of 30 days. Each potential visitor on Google search is assigned to either a treatment group (10%), or a control group (90%).

4.2.1 Treatment Group. 10% of users are assigned to the treatment group. In our experiment, the treatment setting is the use of annotations from the proposed pipeline as the meta tags embedded in the web page source code.

4.2.2 Control Group. 90% of users remain in the control group. In our experimental design, the control is the use of the existing meta tags. These meta tags are produced using broad heuristics to summarize the text of a page. Specifically, text data is collected across all pins on a board. Each annotation is then scored by tf-idf. The top three annotations by tf-idf are used as the meta-tags. For example, most existing meta tags take the structure, "`number_of_pins` best pictures of `topic`".

4.2.3 Evaluation. We evaluate the deployed pipeline (treatment) using the metrics listed below. All differences are calculated as difference = treatment - control.

- **`sess. diff`** The percentage difference in session activity directed from Google search. This is computed across all device types.
- **`crawled sess. diff`** The percentage of unique pages that have received traffic from Google and have been crawled by Googlebot. This is computed across all device types.
- **`sess. diff (desktop)`** The percentage of unique pages that have received traffic from Google and have been crawled by Googlebot. This is computed across desktop devices only.
- **`sess. diff (mobile)`** The percentage difference in session activity directed from Google search for mobile devices.
- **`crawled sess. diff (desktop)`** The percentage of unique pages that have received traffic from Google and have been crawled by Googlebot. Desktop devices only.
- **`crawled sess. diff (mobile)`** The percentage of unique pages that have received traffic from Google and have been crawled by Googlebot. Mobile devices only.

4.2.4 Results. The results for the online SEO experiment are shown in two tables: results from days 1-15 are shown in table 2 and results from days 16-30 in table 3. In our experimental design, two features ensure that the results are robust to random noise. First, we run the

Features	Accuracy	ROC (AUC)	Precision	Recall
Naive (L-1 score)	0.498 (0.00)	0.500 (0.00)	0.770 (0.05)	0.699 (0.02)
L-1 score, log IDF, term in board	**0.763 (0.01)**	0.845 (0.03)	**0.795 (0.05)**	0.731 (0.02)
L-1 score, log IDF, term in board, term in board * log IDF	0.758 (0.01)	0.843 (0.02)	0.785 (0.04)	0.731 (0.02)
L-1 score, log IDF, term in board, score * log IDF, term in board * log IDF	0.758 (0.01)	0.844 (0.03)	0.777 (0.04)	**0.742 (0.01)**
L-1 score, log IDF, term in board	0.703 (0.07)	0.768 (0.05)	0.724 (0.07)	0.664 (0.10)
L-1 score, log IDF, term in board, L-1 score * log IDF	**0.763 (0.01)**	**0.846 (0.03)**	**0.795 (0.05)**	0.731 (0.02)

Table 1: Results from our model selection procedure using 5-fold cross-validation. Each model is evaluated on pairwise sets of board annotations. The task is to predict which annotation is most relevant to the board in each pair. The means and standard deviations are reported for accuracy, ROC (AUC), precision, and recall. We compare five logistic regression feature sets to a naive baseline of using the L-1 score rank. Including the interaction term "term in board" * log IDF yields small gains in ROC (AUC). We use the feature set from the bottom row in our online experiment.

Table 2: Online SEO experiment results: day 1 - 15. All metrics are percentage differences of the treatment relative to control (e.g., diff = treatment - control). Figures with two stars (e.g., $1\%^{**}$) are statistically significant at the 95% confidence level.

metric	day 1	day 2	day 3	day 4	day 5	day 6	day 7	day 8	day 9	day 10	day 11	day 12	day 13	day 14	day 15
sess. diff	0%	0%	$1\%^{**}$	0%	0%	0%	$1\%^{**}$	0%	0%	0%	$1\%^{**}$	$1\%^{**}$	$1\%^{**}$	1%	0%
crawled sess. diff	0%	0%	$1\%^{**}$	1%	0%	0%	$1\%^{**}$	1%	0%	1%	$1\%^{**}$	$1\%^{**}$	$1\%^{**}$	$1\%^{**}$	1%
sess. diff (desktop)	0%	0%	0%	0%	1%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
sess. diff (mobile)	0%	1%	1%	1%	1%	1%	1%	1%	1%	1%	$1\%^{**}$	1%	1%	1%	1%
crawled sess. diff (desktop)	$-1\%^{**}$	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
crawled sess. diff (mobile)	0%	$1\%^{**}$	$1\%^{**}$	1%	1%	1%	1%	1%	1%	1%	$2\%^{**}$	$2\%^{**}$	1%	$1\%^{**}$	1%

Table 3: Online SEO experiment results: day 16 - 30. All metrics are percentage differences of the treatment relative to control (e.g., diff = treatment - control). Figures with two stars (e.g., $1\%^{**}$) are statistically significant at the 95% confidence level. We observe 15 consecutive days of statistically significant increases in traffic coming to the platform from Google search in the latter half of the experiment.

metric	day 16	day 17	day 18	day 19	day 20	day 21	day 22	day 23	day 24	day 25	day 26	day 27	day 28	day 29	day 30
sess. diff	$1\%^{**}$	$1\%^{**}$	$1\%^{**}$	$1\%^{**}$	$1\%^{**}$	$1\%^{**}$	$1\%^{**}$	$1\%^{**}$	$1\%^{**}$	$1\%^{**}$	$1\%^{**}$	$1\%^{**}$	$1\%^{**}$	$1\%^{**}$	$1\%^{**}$
crawled sess. diff	$1\%^{**}$	$1\%^{**}$	$1\%^{**}$	$1\%^{**}$	$2\%^{**}$	$1\%^{**}$	$1\%^{**}$	$1\%^{**}$	$1\%^{**}$	$1\%^{**}$	$1\%^{**}$	$1\%^{**}$	$1\%^{**}$	$1\%^{**}$	$1\%^{**}$
sess. diff (desktop)	1%	1%	$1\%^{**}$	$1\%^{**}$	1%	1%	1%	1%	0%	0%	1%	$1\%^{**}$	1%	$1\%^{**}$	$1\%^{**}$
sess. diff (mobile)	1%	1%	1%	0%	$1\%^{**}$	0%	0%	1%	0%	1%	0%	0%	0%	0%	0%
crawled sess. diff (desktop)	1%	1%	1%	1%	$1\%^{**}$	1%	0%	1%	1%	0%	0%	$1\%^{**}$	0%	1%	$1\%^{**}$
crawled sess. diff (mobile)	1%	1%	1%	1%	$1\%^{**}$	1%	0%	1%	1%	0%	0%	$1\%^{**}$	0%	1%	$1\%^{**}$

experiment for a full 30 days. Such a long window reduces the risk that changes are due to extraneous factors instead of the treatment. Second, we test each metric for statistical significance at the 95% confidence level. Figures reported with two stars (e.g., $1\%^{**}$) are statistically significant.

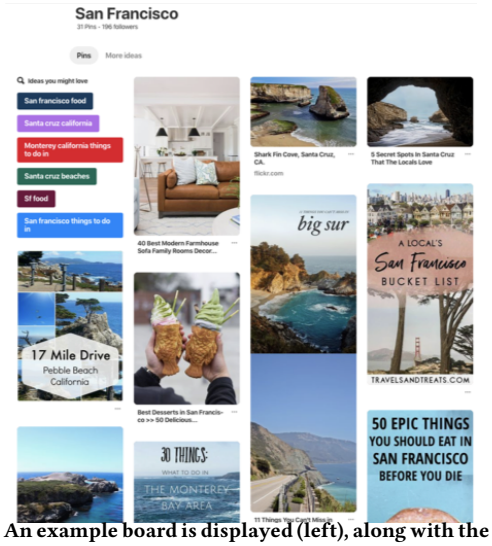
The results from tables 2 and 3 demonstrate that the proposed annotations pipeline causes 1%-2% increase in traffic from Google search. The first two lines of the tables show sessions and page visits across all devices. Both metrics, `sess. diff` and `crawled sess. diff` see generally positive increases over the first 15 days. Both see a statistically significant increase of 1% in 6 of 15 days. During the second half of the experiment we see the best results. For `sess. diff` we see a statistically significant 1% increase in each of the 15 days. We observe even better results for `crawled sess. diff`: 14 of 15 days result in a 1% increase and day 20 sees a 2% increase. All of these results are significant. The other metrics give insight into what type of users are most sensitive to the deployed annotations. In general, traffic from mobile users tends to be higher than desktop.

These results show that the automated pipeline discussed in section 3 for generating natural language content annotations drive an increase in traffic from Google search relative to simpler keyphrases. These results are robust and statistically significant. Moreover, the annotations may attract more users on mobile devices compared to desktop.

4.3 Case Studies

In this section we perform a case study of the proposed annotations pipeline. In particular, we analyze differences in L-1 and L-2 scores relative to the content of a single board. This case study helps us gain insights into why the learning-to-rank model is necessary for producing better content descriptions.

In figure 3 we show content from an actual board, titled "San Francisco" (left). Additionally, we show the top 25 L-1 scores and



L-1 Scores		L-2 Scores	
california travel	0.66	california travel	0.63
california travel road trips	0.52	california travel road trips	0.63
travel usa	0.41	san francisco	0.52
california dreamin'	0.41	san francisco travel	0.50
san fran	0.37	travel usa	0.49
travel	0.35	california dreamin'	0.46
san francisco travel	0.34	san fran	0.45
california dreaming	0.30	west coast road trip	0.44
trip	0.30	california love	0.43
san francisco	0.29	california dreaming	0.43
california	0.26	san francisco with kids	0.41
road trip	0.26	california vacation	0.41
places to travel	0.26	francisco	0.41
california love	0.25	monterey california	0.40
west coast road trip	0.25	breakfast sacramento	0.40
us travel	0.23	san francisco travel guide	0.39
california vacation	0.21	california coast	0.39
california coast	0.20	carmel by the sea	0.39
trek	0.19	carmel california	0.39
future travel	0.18	fall family fun	0.39
travel bugs	0.18	monterey bay aquarium	0.39
monterey	0.18	spring weekend getaway	0.39
monterey bay aquarium	0.17	vernal falls	0.38
places	0.17	san francisco food	0.38
carmel by the sea	0.16	san francisco bucket list	0.38

Figure 3: An example board is displayed (left), along with the L-1 (center) and L-2 annotation scores (right). A green plus means the annotation rank increased from L-1 to L-2, while the red negative indicates the rank decreased. The example board contains pins related to travel tips and tourist sites around San Francisco, CA. The top two annotations for both L-1 and L-2 scores are "california travel" and "california road trips." However, L-1 scores tend to yield more broad, or generic terms such as "travel", "trip", or "trek." Conversely, L-2 scores surface more descriptive terms. such as "west coast road trip" or "san fransisco with kids." The logistic regression ranking layer helps identify more specific and semantically meaningful annotations. These semantic refinements are likely what drives more user traffic from Google search

their annotations (center) as well as the L-2 scores (right) and their corresponding annotations. A green plus means the annotation rank increased from L-1 to L-2, while the red negative indicates the rank decreased. The example board in figure 3 contains pins related to travel tips and tourist sites around San Francisco, CA. We can see that the top two annotations are "california travel" and "california road trips." Both of these terms seem to capture the semantic meaning of the board. We also note that the L-1 scores tend yield more broad, or generic terms such as "travel", "trip", or "trek." Conversely, the L-2 scores surface more descriptive terms. Take for example, "west coast road trip"; this term is ranked 8 places higher in L-2 compared to L-1. Additionally, the L-2 scores capture user-provided semantic information. The term "San Francisco", the user-generated title of the board, is ranked 3rd under the L-2 scheme, and 10th in the L-1 scheme. Finally, the L-2 scores can capture interesting, fine-graned semantic understanding. The term "san francisco with kids" appears in the L-2 top 25, but not in L-1. Further inspection of this board reveals that two or three pins link to articles recommending family friendly sites to see in the Bay Area.

5 RELATED WORK

The current work touches on two major streams of literature: 1) text generation and 2) search engine optimization.

5.0.1 Text Generation. Text generation has been studied for many years by the NLP community. Two main areas in the domain include image captioning, and deep generative modeling. Hu et al. [8] propose a neural generative model based on variational auto-encoders to generate sentences controlled by semantic structure. Many studies seek to generate a descriptive caption given an input image. You et al. [15] propose a recurrent neural network approach that learns text and image semantics through an attention mechanism. Other work proposes learning a multi-modal embedding over images and text to generate captions in an RNN framework [9]

5.0.2 Search Engine Optimization. Search engine optimization research is an important problem in marketing, although formal academic research is somewhat scant. Recent work studies SEO for both sponsored (paid) and organic search results. They find that in the absence of sponsored search, investment in SEO improves a page's organic ranking when the page quality is high [2]. Other work studies consumer response to sponsored search and identifies key features of advertisements that drive click-through rates [7].

6 CONCLUSION

In this paper we proposed a framework for describing content at scale with natural language annotations. These annotations are both descriptive and specific, and also leverage important user-provided context. Such annotations are useful for a variety of purposes; we experimentally show that they can be used as keyword tags for search engine optimization. Specifically, we deploy an extensive online experiment at Pinterest and show that the annotations from the proposed pipeline increases traffic from Google search 1-2% as compared to more generic tags. Moreover, we observe 15 consecutive days of statistically significant results.

ACKNOWLEDGMENTS

We would like to thank James Ouhyoung, Koichiro Narita, and Arun Jay for their invaluable help and feedback on this work.

REFERENCES

[1] [n.d.]. Meta Tags - How Google Meta Tags Impact SEO. *searchengineland.com* ([n. d.]). https://searchengineland.com/guide/what-is-seo
[2] Ron Berman and Zsolt Katona. 2013. The Role of Search Engine Optimization in Search Marketing. *Marketing Science* (July 2013).

[3] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent Dirichlet Allocation. *Journal of Machine Learning Research* (2003).
[4] Chris et al. Burges. 2005. Learning to Rank Using Gradient Descent. In *In proceedings of the International Conference on Machine Learning, ICML'05.*
[5] et al. Carterette. 2008. Here or There: Preference Judgements for Relevance. In *In Proceedings of European Conference on Information Retrieval, ECIR 2008.*
[6] Andew M. Dai, Christopher Olah, and Quoc V. Le. 2015. Document Embedding with Paragraph Vectors. In *ArXiv.* * pages. https://arxiv.org/abs/1507.07998
[7] Anindya Ghose and Sha Yang. 2009. An Empirical Analysis of Search Engine Advertising: Sponsored Search in Electronic Markets. *Management Science* (July 2009).
[8] Zhiting et al. Hu. 2017. Toward Controlled Generation of Text. In *In proceedings of the 34th International Conference on Machine Learning, ICML'17.*
[9] Junhai et al. Mao. 2015. Deep Captioning with Multimodal Recurrent Neural Networks (M-RNN). In *In proceedings of the International Conference of Learning Representations, ICLR'15.*
[10] Kevin P. Murphy. 2012. *Machine Learning: A Probabilistic Perspective.* The MIT Press, Cambridge, Massachusetts.
[11] Jay Adams Paul Covington and Emre Sargin. 8. Deep Neural Networks for YouTube Recommendations. In *RecSys'16 September 15-19, Boston, MA, USA.* ACM, 10 Pages.
[12] Robyn Speer, Joshua Chin, Havasi, and Catherine. 2017. ConceptNet 5.5: An Open Multilingual Graph of General Knowledge. In *In Proceedings of the conference for the Advacement of the Artificial Intelligence, AAAI'17.*
[13] Heath Vinicombe. 2019. Understanding Pins through keyword extraction. *Medium* (July 2019).
[14] Wordstream. [n.d.]. Meta Tags - How Google Meta Tags Impact SEO. *WordStream* ([n. d.]). https://www.wordstream.com/meta-tags
[15] Quanzeng et al. You. 2016. Image Captioning with Semantic Attention. In *In proceedings of the IEEE Conference on Computer Vision and Pattern Recognition CVPR'16.*