Active Search using Meta-Bandits

Shengli Zhu
Lehigh University
zshengli98@gmail.com

Jakob Coles
Lehigh University
jakob.coles@protonmail.com

Sihong Xie
Lehigh University
xiesihong1@gmail.com

ABSTRACT

There are many applications where positive instances are rare but important to identify. For example, in NLP, positive sentences for a given relation are rare in a large corpus. Positive data are more informative for learning in these applications, but before one labels a certain amount of data, it is unknown where to find the rare positives. Since random sampling can lead to significant waste in labeling effort, previous ”active search” methods use a single bandit model to learn about the data distribution (exploration) while sampling from the regions potentially containing more positives (exploitation). Many bandit models are possible and a sub-optimal model reduces labeling efficiency, but the optimal model is unknown before any data are labeled. We propose Meta-AS (Meta Active Search) that uses a meta-bandit to evaluate a set of base bandits and aims to label positive examples efficiently, comparing to the *optimal* base bandit with hindsight. The meta-bandit estimates the mean and variance of the performance of the base bandits, and selects a base bandit to propose what data to label next for exploration or exploitation. The feedback in the labels updates both the base bandits and the meta-bandit for the next round. Meta-AS can accommodate a diverse set of base bandits to explore assumptions about the dataset, without over-committing to a single model before labeling starts. Experiments on five datasets for relation extraction demonstrate that Meta-AS labels positives more efficiently than the base bandits and other bandit selection strategies.

CCS CONCEPTS

• **Information systems → Crowdsourcing**; • **Theory of computation → Sequential decision making**; *Online learning theory*; *Active learning*.

KEYWORDS

Crowdsourcing; active search; bandit

ACM Reference Format:
Shengli Zhu, Jakob Coles, and Sihong Xie. 2020. Active Search using Meta-Bandits . In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management (CIKM '20), October 19–23, 2020, Virtual Event, Ireland.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3340531.3417409

1 INTRODUCTION

In many applications of classification, a large unlabeled dataset can be collected without much cost (e.g. via Internet crawling), while identifying the rare positive data instances can be labor-intensive. In NLP, relation extraction classifies a pair of words/phrases in a sentence into one of the multiple relations. For example, a model can extract the relation *hasSpouse* between the occurrences of the two concepts (*Barack_Obama*, *Michelle_Obama*) from the text snippets "... *Barack Obama and his wife Michelle Obama* ...". Such an occurrence of the pair in a sentence is called a *mention* or an instance, which needs to be classified as having a specific relation (positive) or not (negative). The current best relation extractors are based on machine learning and need to be trained on a large number of labeled data. And yet humans have to label the data, which is costly. In particular, positive mentions are informative for model training and yet are much more rare and difficult to find. The more prevalent negative mentions can be obtained less expensively by randomly sampling pairs of arbitrary words/phrases from any sentences.

To learn a relation extractor on a new corpus, one starts with an unlabeled dataset. Weak supervision for relation extraction [20] can be helpful in data augmentation but can only complement human-labeled data whenever an annotation budget is available. Data annotation will face the exploration *vs.* exploitation dilemma: an annotator needs to find positive samples without knowing where they are ahead of time, but also has to explore the data distribution by labeling certain samples that are negatives. On the one hand, more exploratory labeling leads to better knowledge of the data distribution but leaves less budget to exploit the regions that have more positives. On the other hand, less exploration may trap the annotator in a narrow part of the dataset, missing the areas potentially with more positives.

To address the dilemma, active search (AS) algorithms are designed to carefully balance the exploration and exploitation in order to obtain the most positive samples. Active search differs from active learning (AL): an AL algorithm labels data for a specific model and is evaluated by the model's performance trained on the data, while an AS algorithm is model-agnostic and aims to find as many positives as possible that potentially can be used to train a model, construct a knowledge base, or for other downstream tasks. Greedy active search algorithms with limited steps of look-ahead and without exploration have been proposed in [6, 8, 23], but their search scales poorly with the number of unlabeled data due to the look-ahead. Bandit algorithms can manage the trade-off between exploration and exploitation in active search [2–4, 11, 12, 17, 18]. However, prior bandit based active search [13, 16] committed to a single bandit model by making an implicit assumption about the data distribution, which is unknown before the bandit is selected. As a result, the selected bandit can be sub-optimal with respect to the true data distribution. Indeed, we empirically show that different bandit algorithms lead to a rather diverse annotation efficiency. One possible explanations is that, a bandit may assume that the positives are from multiple clusters under a specific distance

CIKM '20, October 19–23, 2020, Virtual Event, Ireland

ACM ISBN 978-1-4503-6859-9/20/10...$15.00
https://doi.org/10.1145/3340531.3417409

metric, while another bandit uses a different distance metric, so that the positives form a single cluster and less exploration is needed.

To avoid over-committing to a presupposition of data distribution of a particular dataset, in an online fashion, the "usefulness" of different bandit algorithms should be actively evaluated (exploration of the algorithms), and then be selected to propose unlabeled data for annotation (exploration in the data space). For the purpose, we propose Meta-AS (Meta Active Search) that has the following benefit over the prior work. First, Meta-AS has a set of more diverse bandit models, including linear [3, 11, 17], kernel-based [18], and graph-based bandits [1, 21] to fit the unknown data distribution. The EXP4 algorithm [2] uses a bandit to manage a set of linear bandits, while Meta-AS adopts more diverse bandits to allow more exploration in the algorithm space. Second, specific to the annotations for relation extraction, a bag of multiple mentions regarding the same pair of words/phrases appearing in one or multiple sentences, can be annotated all at once to save human mental effort. We design Meta-AS to propose bags of mentions, rather than individual mentions.

2 PROBLEM DEFINITION

Let $\{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ denote the feature vectors of n mentions, which are initially in the set of unlabeled data $\mathcal{U} = \{\mathbf{x}_i\}_{i=1}^n$. Let $B^j = \{\mathbf{x}_i^j, i = 1, \ldots, m_j\} \subset \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ denote the j-th bag of mentions (instances). The label of $\mathbf{x}_i$ is $y_i \in \{0, 1\}$. A bag is positive if and only if at least one of the mentions it contains is positive. The subset $\mathcal{P} \subset \mathcal{U}$ is the designated set of positive instances to be discovered using a budget of M units of human annotation. The generic active search algorithm works iteratively, as shown in Algorithm 1. At iteration t, the algorithm chooses a bag $B^{j(t)}$ from the unlabeled set $\mathcal{U}$ and asks the human annotator to label the mentions in $B^{j(t)}$. Since we are interested in collecting positive data, an instance labeled as positive leads to reward 1 and a negative instance leads to reward 0. The model is updated according to the label of instances in $B^{j(t)}$, which are then removed from $\mathcal{U}$ as their labels are now revealed. The labels of the data points that are not selected remain unknown. The goal is to design a query strategy, parametrized by $\boldsymbol{\theta}$, that can select $B^{j(t)}$ over $t = 1, \ldots$ until the budget is used up, so that the number of positive instances labeled is maximized.

Algorithm 1 Generic Active Search

```
Input: unlabeled dataset $\mathcal{U}$; labeling budget $M$.
Output: labeled dataset $\mathcal{L}$
Initialize the parameters $\boldsymbol{\theta}$ of the selection strategy.
$t = 1$; $\mathcal{L}^{(t)} = \emptyset$.
for $t = 1, \ldots$ do
    if $|\mathcal{L}^{(t)}| \geq M$ then
        Out of budget and return $\mathcal{L}^{(t)}$.
    end if
    $(*)$ Use $\boldsymbol{\theta}$ to select a bag $B^{j(t)}$ of unlabeled instances.
    Label the instances in $B^{j(t)}$.
    Remove $B^{j(t)}$ from $\mathcal{U}$ and let $\mathcal{L}^{(t+1)} = \mathcal{L}^{(t)} \cup B^{j(t)}$.
    Update parameters $\boldsymbol{\theta}$ using the labeled data in $B^{j(t)}$.
end for
```

3 METHODOLOGIES

The above generic framework can be implemented using a single bandit algorithm, such as UCB [3] or Thompson sampling [4, 11], or a meta-bandit with multiple base bandit algorithms. We describe each of these bandit algorithms.

3.1 Base bandits

3.1.1 UCB. The UCB [3] for the multi-armed bandit problem aims to attain maximal cumulative rewards by pulling K arms $\{1, \ldots, K\}$ over time to explore the arm reward distributions and collect the rewards. At iteration t, it pulls the arm $a(t) \in \{1, \ldots, K\}$ with the highest upper bound of the mean rewards:

$$a(t) = \underset{j \in \{1, \ldots, K\}}{\operatorname{argmax}} \ \bar{\mu}_j(t) + C\sqrt{\frac{2 \ln t}{t_j}} \tag{1}$$

where $\bar{\mu}_j(t)$ is the empirical average reward collected by arm j up to time t, and t_j is the number of times the arm j has been pulled up to time t. The parameter $\boldsymbol{\theta}$ in Algorithm 1 for UCB is $\bar{\mu}_j(t)$ and t_j for $j = 1, \ldots, K$. When the reward of the pulled arm $a(t)$ is revealed, $\bar{\mu}_j(t)$ and t_j will be updated accordingly for the next iteration.

UCB will be used in two ways. As a base bandit, the data will be clustered into K clusters, so that each resulting cluster is an arm. See Section 3.1.4 for more details. UCB will be used as a meta-bandit to select base bandits as arms (see Section 3.2).

3.1.2 Thompson sampling. Thompson sampling (TS for short) is a Bayesian treatment of the bandit problem. Here we adopt Thompson sampling in the contextual bandit setting [4] to take feature vectors $\mathbf{x}_i$ into account. As the rewards are binary, logistic regression is used to model the likelihood of a positive label/reward $p(y_i = 1|\mathbf{x}_i; \boldsymbol{\theta})$, given a context $\mathbf{x}_i$ and a linear model $\boldsymbol{\theta}$:

$$p(y_i = 1|\mathbf{x}_i) = (1 + \exp(-\mathbf{w}^\top \mathbf{x}_i))^{-1}. \tag{2}$$

For exploration, TS draws a sample of $\mathbf{w}$ from the posterior $p(\mathbf{w}|\mathcal{L}^{(t)})$, where $\mathcal{L}^{(t)}$ is the set of instances labeled up to time t. Using Laplace approximation, the posterior $p(\mathbf{w}|\mathcal{L}^{(t)})$ can be parametrized by the mean vector $\mathbf{m}$ and a diagonal covariance matrix $\mathbf{q}$. When a new batch of data is labeled, the mean and the covariance are updated in an online fashion. See [4] for more details.

3.1.3 UCB using Gaussian Process. TS assumes a linear relationship between y_i and $\mathbf{x}_i$. The GP-UCB algorithm [18] uses Gaussian Process and kernel functions to model a nonlinear relationship between the data and the labels. Let $f(\mathbf{x})$ be a function that predicts the reward when $\mathbf{x}$ is selected for labeling. GP-UCB assumes the function f is sampled from a Gaussian Process $\mathbf{GP}(f)$, which controls the smoothness of f via a kernel function $k(\mathbf{x}, \mathbf{x}') \in \mathbb{R}$ for any two instances $\mathbf{x}$ and $\mathbf{x}'$. Given ℓ labeled points $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_\ell, y_\ell)\}$, the reward of any unlabeled point $\mathbf{x}$ can be estimated using the posterior Gaussian distribution with the following mean and variance:

$$\begin{aligned} f_\ell(\mathbf{x}) &= \mathbf{k}_\ell(\mathbf{x})^\top (K_\ell + \sigma^2 I_\ell)^{-1} \mathbf{y}_\ell, \\ \sigma_\ell^2(\mathbf{x}) &= k(\mathbf{x}, \mathbf{x}) - \mathbf{k}_\ell(\mathbf{x})^\top (K_\ell + \sigma^2 I_\ell)^{-1} \mathbf{k}_\ell(\mathbf{x}), \end{aligned}$$

where $\mathbf{k}_\ell(\mathbf{x}) = [k(\mathbf{x}, \mathbf{x}_1), \ldots, k(\mathbf{x}, \mathbf{x}_t)]^\top$ is the function $k(\mathbf{x}, \cdot)$ evaluated on the ℓ labeled instances, K_ℓ is the kernel matrix with $K_\ell(i, j) =$

$k(\mathbf{x}_i, \mathbf{x}_j)$, and $\mathbf{y}_\ell = [y_1, \ldots, y_\ell]^\top$. GP-UCB [18] selects

$$\mathbf{x} = \operatorname{argmax}_{\mathbf{x} \in \mathcal{U}} \left[f_\ell(\mathbf{x}) + \beta^{1/2} \sigma_\ell(\mathbf{x}) \right], \tag{3}$$

where β balances exploitation and exploration.

3.1.4 *UCB using graph information.*

If there are some relationships among the instances beyond the data vectors, a graph can describe the relationships to benefit active search. In the graph, an instance is a node and two related nodes are connected by an edge. For example, the "homophily relationship" assumes that connected instances are more likely to have similar labels. In particular, graph bandits (Graph-UCB for short) assume that the rewards of a node can be inferred from those of its neighbors [1, 21]. We construct a graph of mentions so that two mentions are connected if they shared at least one word. The motivation is that if a mention is positive, then other mentions including the shared word(s) are likely to be positive. We first run DeepWalk [15] on the constructed graph to obtain embeddings of the nodes, and then run the k-means algorithm to cluster the nodes into clusters. Each cluster will contain instances that are likely to have similar labels. A UCB bandit uses the clusters as arms and all instances from a cluster will receive the same UCB score (Eq. (1)).

3.1.5 *Proposing Bags of Instances.*

Each base bandit (TS, GP-UCB, Graph-UCB) returns a score for each instance while we need to propose multiple instances in a bag to reduce annotator mental effort. For each bag and each base bandit, we can take the minimum, average, or maximum of the scores of instances in the bag to estimate the value of the bag. Note that different instances from a bag can belong to multiple clusters in Graph-UCB.

3.2 Meta-bandit for Active Search

We don't know which of the above bandits (called "base" bandits) is the best for active search on an unlabeled dataset. We propose Meta-AS, a "meta-bandit" that uses UCB to learn to select base bandit to propose data for labeling to collect as many positives as possible. The faster Meta-AS can find the optimal base bandit, the more remaining budget can be used to exploit the positive part of the dataset. The meta-bandit maintains performance statistics of a set $\mathcal{A}$ of base bandit algorithms. To diversify the base bandit portfolio, we set different values to the hyper-parameters of each base bandit (kernel function, exploration rate, etc.). At iteration t of Algorithm 1, step $(*)$, Meta-AS first samples a base bandit according to

$$a(t) = \operatorname{argmax}_{a \in \mathcal{A}} \left\{ \frac{p_a(t)}{p(t) \times r_a(t)} + C\sqrt{\frac{\ln t}{n_a(t)}} \right\}, \tag{4}$$

where $p_a(t)$ is the number of positives collected using the a-th base bandit, $p(t) = \sum_a p_a(t)$ is the number of all positive instances collected so far, and $r_a(t)$ is the proportion of iterations that arm a was selected by the meta-bandit among the first t iterations. The exploration is controlled by the hyper-parameter C and how frequent the base bandit a has been used to sample data ($n_a(t)$). The selected base bandit (indexed by $a(t) \in \mathcal{A}$) then selects an unlabeled bag $B^{j(t)}$ for labeling, and the labeled instances are used to update the parameters of the selected base bandit (e.g. Graph-UCB) and the parameters $p_a(t)$, $p(t)$, $r_a(t)$ and $n_a(t)$ of the meta-bandit (Eq. (4)).

Table 1: Datasets statistics

Datasets	# of Bags	# of Pos Instances	# of all instances	# Edges
Pet	1163	721	3052	204883
Auto	1439	776	2758	99708
Instruments	1463	1084	3946	343214
Videos	1963	841	3841	339603
5-Products	961	795	2117	118309

4 EXPERIMENTS

Datasets. We adopt a review corpus used in [7] and four larger product review corpora (Pet, Auto, Instruments, and Videos) from [14]. Our goal is to discover mentions of an adjective and a noun that appear in the same sentence, where the adjective modifies the noun (considering "modification" as a relationship). For example, in the sentence "*This large screen is what I have been looking for for long time.*", the adjective "*large*" modifies "*screen*" and they are a positive mention of the pair ("*large*", "*screen*"), while the pair ("*long*", "*screen*") is a negative one. The statistics of the bags and instances in the corpora are listed in Table 1.

Bandits settings and baselines. There are hyper-parameters for the base bandits: the covariance matrix $\mathbf{q}$ in Thompson sampling (set to the identity matrix multipled by a scalar in $\{0, 0.01, 0.1, 1.0, 10.0\}$), the bandwidth of the radial basis function kernel in GP-UCB (set to the inverse of the number of features in $\mathbf{x}$), the β in Eq. (3) (set to values from $\{1, 10, 100, 1000, 10000\}$). We vary the length of walks in $\{3, 4, 7, 10\}$ with a fixed walk length 80 for DeepWalk and the number of clusters in Graph-UCB is fixed at 20 for all datasets. When running UCB on the resulting clusters, the hyper-parameter $C = \sqrt{2}$ in Eq. (1). To show that Meta-AS can learn from a set of base bandits with diverse performances, on each unlabeled dataset, we evaluate each base bandit with different combinations of hyper-parameter values and aggregation functions (min, mean, and max) for bag proposal, and identify the worst, median, and best settings. We then have three instances of TS, denoted as TS-0, TS-1, and TS-2, with increasing performances, and similarly for GP-UCB (GP-0, GP-1, and GP-2) and Graph-UCB (GU-0, GU-1, and GU-2). Note that GP-UCB is not scale only tested on the small dataset.

We use two sets of baselines. The first contains any single base bandit with a particular hyper-parameter setting. By comparing the performances of Meta-AS and a single base bandit, we can see how much Meta-AS can approach or even exceed the best base bandit, and by comparing different base bandits with different hyper-parameters, we can see the diverse performance of the base bandits and confirming the need of bandit selection during data annotation.

4.1 Results

Overall performance. We run Meta-AS with the selected set of base bandits on the five datasets. In Figure 1, we show the performance of each bandit, measured in recall (the percentage of all positive instances that are selected and annotated by the bandit). On the smaller 5-product dataset, Meta-AS is the runner-up among all nine base-bandits, indicating that Meta-AS learned to identify the best base bandit and aimed to approximate the best performance. Even more interesting is the performance of Meta-AS on the four larger corpora: Meta-AS learned to find the best base bandit and started to exceed the best after between 200 to 300 rounds of human feedback. We believe that the better performance comes from less exploration

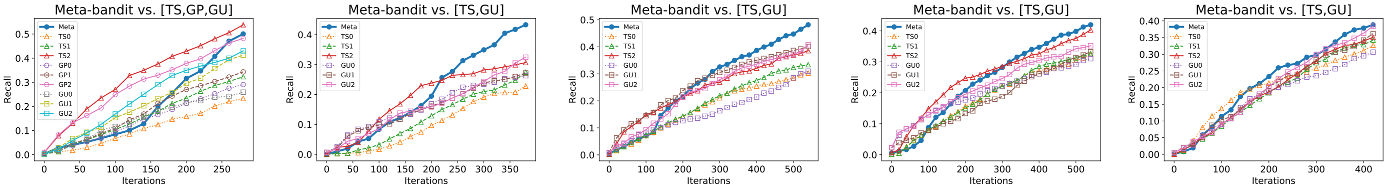

Figure 1: From left to right: how the recall rates change as Meta-AS and the base bandits on the datasets 5-Products, Pet, Auto, Instruments, and Videos.

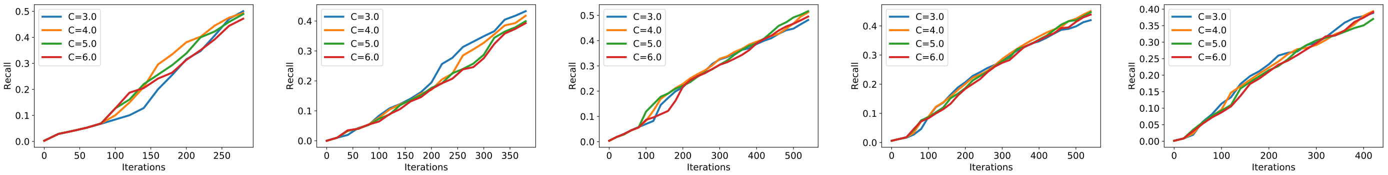

Figure 2: From left to right: the performance of Meta-AS under different exploration parameter C on the five datasets.

in the meta-bandit, which safely enter the exploitation mode early. Note the diversity of the performances in the base bandits indicate the necessity of Meta-AS to learn to avoid the low-performing bandits. Although not shown here, we observed that more base bandits leads to better Meta-AS performance, confirming the capability of Meta-AS to learn to focus on the optimal base bandit.

Sensitivity studies. In Figure 2, we study how sensitive Meta-AS is to its hyper-parameter C in Eq. (4) that controls the amount of exploration in the space of base bandits. We can see that on all five datasets, the hyper-parameter C (set to values in $\{3, 4, 5, 6\}$) does not affect the performance of Meta-AS much.

5 RELATED WORK AND CONCLUSION

The selection of base bandit algorithms is an instance of "algorithm selection" [5] and more broadly an instance of "meta learning" [22]. In [5], they proposed to use a bandit to select SAT solvers to find solutions to different SAT problem instances in an online learning setting. In meta learning [22], features describing different problem instances (in our case, different datasets to be actively searched) are designed to guide the selection the best from a portfolio of algorithms to solve each problem. In [9], the performance of a trained classifier is actively evaluated. An ensemble of base bandits has been applied to recommendation systems [24]. However, the experts or base models themselves are not learning as more data are annotated and therefore can't reflect the latest exploration results.

We conclude that Meta-AS is needed when multiple search algorithms exist but have unknown and diverse performance. We plan to include more base bandit algorithms in Meta-AS, and demonstrate its usefulness beyond text data.

ACKNOWLEDGMENT

This work is supported by NSF through grants CNS-1931042, IIS-2008155, and CNS-1757787. We would like to thank Sam Chebruch for implementing the GP-UCB algorithm, and Sheldon Xu for implementing the contextual Thompson sampling.

REFERENCES

[1] N Alon, N Cesa-Bianchi, C Gentile, S Mannor, Y Mansour, and O Shamir. Non-stochastic Multi-Armed Bandits with Graph-Structured Feedback. *SIAM Journal on Computing*, 46(6):1785–1826, 2017.

[2] P Auer, N Cesa-Bianchi, Y Freund, and R Schapire. The Nonstochastic Multiarmed Bandit Problem. *SIAM Journal on Computing*.

[3] Peter Auer, Nicolò Cesa-Bianchi, and Paul Fischer. Finite-time Analysis of the Multiarmed Bandit Problem. *Machine Learning*, 2002.

[4] Olivier Chapelle and Lihong Li. An Empirical Evaluation of Thompson Sampling. NIPS, 2011.

[5] Matteo Gagliolo and Jürgen Schmidhuber. Algorithm Selection as a Bandit Problem with Unbounded Losses. In *Learning and Intelligent Optimization*, 2010.

[6] Roman Garnett, Yamuna Krishnamurthy, Xuehan Xiong, Jeff G Schneider, and Richard P Mann. Bayesian Optimal Active Search and Surveying. In *ICML*, 2012.

[7] Minqing Hu and Bing Liu. Mining Opinion Features in Customer Reviews. AAAI, 2004.

[8] Shali Jiang, Gustavo Malkomes, Geoff Converse, Alyssa Shofner, Benjamin Moseley, and Roman Garnett. Efficient nonmyopic active search. In *Proceedings of the 34th International Conference on Machine Learning*, 2017.

[9] N Katariya, A Iyer, and S Sarawagi. Active Evaluation of Classifiers on Large Datasets. In *ICDM*, 2012.

[10] Andreas Krause and Cheng S Ong. Contextual Gaussian Process Bandit Optimization. In *NIPS*. 2011.

[11] Lihong Li, Wei Chu, John Langford, and Robert E Schapire. A Contextual-bandit Approach to Personalized News Article Recommendation. In *WWW*, 2010.

[12] Lihong Li, Wei Chu, John Langford, and Xuanhui Wang. Unbiased Offline Evaluation of Contextual-bandit-based News Article Recommendation Algorithms. In *WSDM*, pages 297–306, 2011.

[13] Yifei Ma, Tzu-Kuo Huang, and Jeff G Schneider. Active Search and Bandits on Graphs using Sigma-Optimality. In *UAI*, 2015.

[14] Jianmo Ni, Jiacheng Li, and Julian McAuley. Justifying Recommendations using Distantly-Labeled Reviews and Fine-Grained Aspects. In *EMNLP-IJCNLP*, 2019.

[15] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. 2014.

[16] Jean-Michel Renders. Active search for high recall: A non-stationary extension of thompson sampling. In Gabriella Pasi, Benjamin Piwowarski, Leif Azzopardi, and Allan Hanbury, editors, *ECIR*, 2018.

[17] Steven L Scott. A Modern Bayesian Look at the Multi-armed Bandit. *Appl. Stoch. Model. Bus. Ind.*, 26(6):639–658, 2010.

[18] N Srinivas, A Krause, S M Kakade, and M W Seeger. Information-Theoretic Regret Bounds for Gaussian Process Optimization in the Bandit Setting. *IEEE Transactions on Information Theory*, 2012.

[19] Niranjan Srinivas, Andreas Krause, Sham Kakade, and Matthias Seeger. Gaussian Process Optimization in the Bandit Setting: No Regret and Experimental Design. In *ICML*, 2010.

[20] Mihai Surdeanu, Julie Tibshirani, Ramesh Nallapati, and Christopher D Manning. Multi-instance Multi-label Learning for Relation Extraction. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, 2012.

[21] Michal Valko. *Bandits on graphs and structures.* PhD thesis, 2016.

[22] Ricardo Vilalta and Youssef Drissi. A Perspective View and Survey of Meta-Learning. *Artificial Intelligence Review*, 18(2):77–95, 2002.

[23] Xuezhi Wang, Roman Garnett, and Jeff Schneider. Active Search on Graphs. In *KDD*, 2013.

[24] Qingyun Wu, Huazheng Wang, Yanen Li, and Hongning Wang. Dynamic Ensemble of Contextual Bandits to Satisfy Users' Changing Interests. In *WWW*, 2019.