MArBLE: Hierarchical Multi-Armed Bandits for Human-in-the-Loop Set Expansion

Muntasir Wahed
Department of Computer Science
Virginia Tech
Virginia, USA
mwahed@vt.edu

Daniel Gruhl
IBM Research
Almaden Lab
California, USA
dgruhl@ibm.com

Ismini Lourentzou
Department of Computer Science
Virginia Tech
Virginia, USA
ilourentzou@vt.edu

ABSTRACT

The modern-day research community has an embarrassment of riches regarding pre-trained AI models. Even for a simple task such as lexicon set expansion, where an AI model suggests new entities to add to a predefined seed set of entities, thousands of models are available. However, deciding which model to use for a given set expansion task is non-trivial. In hindsight, some models can be 'off topic' for specific set expansion tasks, while others might work well initially but quickly exhaust what they have to offer. Additionally, certain models may require more careful priming in the form of samples or feedback before being finetuned to the task at hand. In this work, we frame this model selection as a sequential non-stationary problem, where there exist a large number of diverse pre-trained models that may or may not fit a task at hand, and an expert is shown one suggestion at a time to include in the set or not, *i.e.*, accept or reject the suggestion. The goal is to expand the list with the most entities as quickly as possible. We introduce MArBLE, a hierarchical multi-armed bandit method for this task, and two strategies designed to address cold-start problems. Experimental results on three set expansion tasks demonstrate MArBLE's effectiveness compared to baselines.

CCS CONCEPTS

• **Information systems** → **Information extraction**; *Thesauri.*

KEYWORDS

Entity Set Expansion, Human-in-the-Loop Set Expansion, Hierarchical Multi-Armed Bandits, Model Selection, Cold-start Problems

ACM Reference Format:
Muntasir Wahed, Daniel Gruhl, and Ismini Lourentzou. 2023. MArBLE: Hierarchical Multi-Armed Bandits for Human-in-the-Loop Set Expansion. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management (CIKM '23), October 21–25, 2023, Birmingham, United Kingdom.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3583780.3615485

1 INTRODUCTION

Entity set expansion starts with a small set of seed entities and aims at expanding the set by including entities with similar semantics.

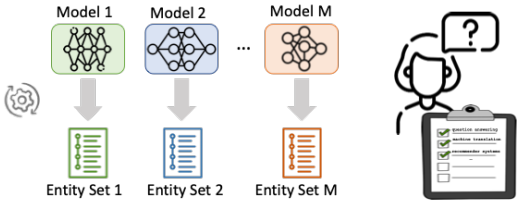


Figure 1: In traditional set expansion, a subject-matter expert has to first select a model. In real-world settings, this is a non-trivial decision, as each model may produce entity sets with varying semantics and the expert must know about the training distribution of the models beforehand. Instead, MArBLE collects candidate entities from a large set of models simultaneously and progressively learns which model to query in each round.

The extracted entities can be useful for various natural language processing (NLP) downstream tasks, such as opinion mining [42], semantic search [6], taxonomy construction [32], query understanding [20, 36], question answering [35], and recommendation [17]. Depending on the downstream task, target semantics for a set expansion task can be general or nuanced. For example, starting with a seed of "`question answering`" and "`machine translation`", the goal can be to extract entities encompassing the different areas of Computer Science research, such as "`image classification`", "`activity detection`", and "`recommender systems`". On the other hand, the goal can be to extract NLP research tasks, such as "`sentiment classification`", and "`text generation`", *etc.*

In practice, there exist several challenges for set expansion methods. There is a vast number of multifaceted data sources in the real world. Additionally, there exists a wide range of methods suitable for set expansion tasks. Each method has distinct unpredictable advantages and limitations that may be observed in retrospect. Exposing the methods to different data sources also dictates the quality of the trained models. For example, even if there is no pre-trained model for a particular domain or task, there may be models trained on related data collections and domains. Being able to capture inherent similarities between such models allows for the retrieval of the most suitable methods. Yet, in many cases, choosing the most suitable method for a given set expansion task is not straightforward. Additionally, what kinds of data collections were used to train a model, and what kind of useful data subsets each training set may contain, are model properties that often remain unknown to users. Therefore, it is beneficial to develop post hoc frameworks that can select among multiple methods based on user feedback.

CIKM '23, October 21–25, 2023, Birmingham, United Kingdom

ACM ISBN 979-8-4007-0124-5/23/10.
https://doi.org/10.1145/3583780.3615485

There has been some work on selecting between multiple methods for other domains, such as recommender systems [10], digital advertising [5, 7], forecasting [12, 46], *etc.* However, the task of human-in-the-loop entity set expansion is uniquely challenging compared to the aforementioned tasks as there is no predetermined objective function that can guide the training process for entity set expansion. Instead, the feedback must be collected from a subject-matter expert at each iteration, and hence the model needs to evolve based on iterative refinements. Additionally, each new entity task may be inherently different from previous ones. A model that worked well for expanding a set of computer science terms might not work well for expanding a set of terms related to foods. Consequently, it is important to incorporate user feedback into the framework to guide the method selection process. Though there has been some work on human-in-the-loop set expansion [1, 13], these methods focus on pre-selecting a single data source and a single method for set expansion. However, given the challenges above, restricting the solution to a single model or dataset is suboptimal in many cases. This line of research has received limited attention.

In this work, we frame the task of entity set expansion as a multi-source human-in-the-loop setting (see Figure 1), and introduce Hierarchical **M**ulti-**Ar**med **B**andit for **L**exicon **E**xpansion (**MArBLE**). In contrast to previous works where it is required to select a model before starting the set expansion task, MArBLE can select from multiple models without making any assumptions on the architectures or training data distributions of the available models. MArBLE can gradually improve its selection based on subject-matter expert feedback. Specifically, the subject-matter expert decides whether to accept or reject entity candidates and MArBLE learns based on this type of feedback. While selecting a model is necessary for traditional approaches, which require knowledge about the architecture and the training distribution of the models, MArBLE removes this pre-requisite, making the process more streamlined, faster, and less expensive. Furthermore, we present two boosting strategies to address the cold-start problem [19], where models may not have sufficient data to provide good candidates in the initial rounds.

The contributions of our work are summarized as follows: **1)** We introduce MArBLE, a human-in-the-loop entity set expansion framework that, based on expert feedback, can select the most suitable model for the task at hand among a large number of available set expansion models. **2)** To alleviate cold-start problems, we propose two boosting methods that utilize collective information from all models to make better-informed decisions in the initial stages. **3)** We validate the efficacy of MArBLE through extensive experiments across 3 benchmark datasets and 81 set expansion models. Experiments show that MArBLE, along with its variants, can successfully identify useful models for various set expansion tasks.

2 RELATED WORK

Entity Set Expansion: Ghahramani and Heller [14] formulate set expansion as a Bayesian inference problem, comparing the posterior probability of an entity given the seed entities to the prior probability of that entity. A few other early approaches rely on co-occurrence statistics [26] and topic information [25] to rank candidate entities. Pantel et al. [23] use distributional semantics to efficiently calculate the pairwise similarity of all terms in a web corpus and later use it for set expansion. Wang and Cohen [33, 34] build a graph by automatically extracting entities from semi-structured webpages with lists and then use a ranking algorithm to find the most promising entities. Other methods similarly exploit knowledge graphs built on web corpora to extend entity sets using meta-paths [29, 45], deficiency (incompleteness) [6, 43], or variational autoencoders [24]. There also exist approaches that incorporate infrequent synonyms into the set [28], probe language models to generate pseudo-class labels [44], apply contrastive learning to finetune language models for the entity set expansion tasks [18], or perform pattern matching (bootstrapping) [8, 39–41]. However, none of these works consider human-in-the-loop settings.

Human-in-the-Loop Entity Set Expansion: A major drawback of the aforementioned methods is that they cannot incorporate subject-matter expert feedback in their architectures. Such feedback can help prevent semantic drift and focus the set expansion task on the target granularity. While there has been some work on incorporating user feedback in model design and label acquisition in various downstream tasks [4, 21, 22, 37, 38], set expansion has received limited attention. Coden et al. [8] generate patterns from seed entities to find similar terms in the corpus and validate the candidate terms with the help of a subject matter expert in each iteration. Gentile et al. [13] propose a human-in-the-loop "explore and exploit" (EnE) paradigm, where the first step is to find similar terms in the corpus by using neural language models (explore phase) and later on generate more terms by employing modifications and extensions (exploit phase). Kohita et al. [16] extract terms based on a weighted similarity function and optimize the process with user feedback. However, all these works are designed with specific models in mind and are not generalizable to handling multiple models. In contrast, our work can automatically select between a large set of available models and adjust based on user feedback.

3 PROPOSED METHOD

3.1 Problem Statement

Given an initial set of seed entities $\mathcal{E} = \{e_1 \dots e_N\}$, the goal is to expand the set with more entities with similar semantics. These latent semantics are defined by a subject-matter expert and are not given a priori. Let $\mathcal{M} = \{m_1 \dots m_M\}$ be the set of entity set expansion models that are available for the task. Each model is denoted as $C = m_i(\mathcal{P}^+, \mathcal{N}^-)$, where $m_i \in \mathcal{M}$ takes as input a set of positive entities $\mathcal{P}^+$ and a set of negative entities $\mathcal{N}^-$, and returns a set of candidate entities $C = \{e_i\}_{i=1}^{|C|}$. By considering both positive and negative samples, the model can focus more quickly on a specific density region of the hypothesis space. Let O be a perfect oracle, *i.e.*, a subject matter expert that provides feedback $f(e_i) \in \{0, 1\},\ \forall e_i \in C$, where 0 and 1 represents negative and positive feedback, respectively, for each candidate $e_i \in C$.

Each model in $\mathcal{M}$ can have different characteristics, *e.g.*, model architecture, model capacity, and training distributions differ, making each model suitable for different tasks. It is thus challenging to select between the set of available models without knowing the underlying model attributes and the training distribution. Even building an ensemble of models is non-trivial in this case, as the set expansion task can be very specific, making it impossible to know

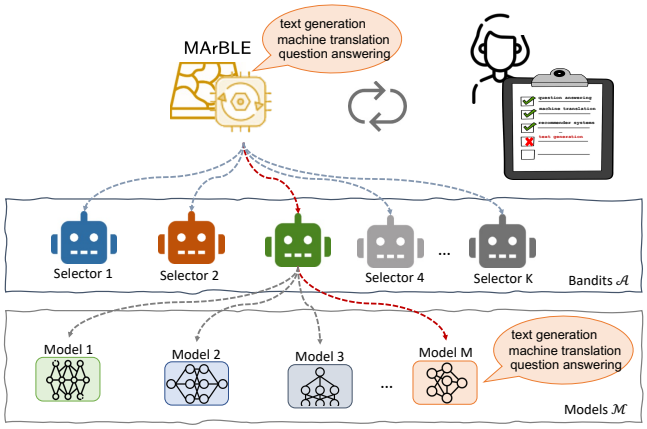


Figure 2: MArBLE is a meta-learner that selects between a set of multi-armed bandit agents $\mathcal{A}$. The selected multi-armed bandit agent $\mathcal{A}_i$ in turn selects between a set of available set expansion models $\mathcal{M}$. Then, the selected model $\mathcal{M}_j$ outputs a set of candidate entities C, which are passed to the subject-matter expert to accept or reject and update the global sets of positive and negative entities $\mathcal{P}^+$ and $\mathcal{N}^-$, respectively.

which models may be relevant. Consequently, it is beneficial to construct a framework that can dynamically improve model selection based on task feedback.

3.2 Sequential Model Selection

Intuitively, model selection can be framed as a multi-armed bandit problem, where each model is an arm and the subject-matter expert feedback is the reward for the corresponding arms. At each time step t, the multi-armed bandit agent A_j first selects a model m_t based on a probability vector $t \sim p_t \in [0,1]^M$, initialized uniformly. The selected model m_t returns a set of candidate entities C_t and the subject matter expert provides feedback for each candidate entity $f_t(e_i), \forall e_i \in C_t$, *i.e.*, whether to accept $(f_t(e_i)=1)$ or reject $(f_t(e_i)=0)$ a candidate e_i[1]. The accepted or rejected entities form subsets $\mathcal{P}_t^+ = \{e_i \in C_t : f_t(e_i)=1\}$ and $\mathcal{N}_t^- = \{e_i \in C_t : f_t(e_i)=0\}$, respectively, which are used to update the global positive and negative sets, *i.e.*, $\mathcal{P}^+ = \mathcal{P}^+ \cup \mathcal{P}_t^+$, and similarly for $\mathcal{N}^-$. We reward model m_t for each entity candidate, *i.e.*, $\mathcal{R}_t(m_t, C_t) = |\mathcal{P}_t^+| - |\mathcal{N}_t^-|$. The process continues with the global set of positives $\mathcal{P}^+$ and negatives $\mathcal{N}^-$ fed to the model m_{t+1} on the next round.

To balance exploitation (select a model that performed well in previous rounds) and exploration (explore the space of models), there exist several ways to update the probability vector p_t based on the reward $\mathcal{R}_t(m_t, C_t)$. For a given model m_t, bandit algorithms compute the distribution $p_t(m_t)$ such that it maximizes the long-term rewards. There are several existing methods to model $p_t(m_t)$. For example, Boltzmann exploration [30] uses a softmax over the reward history to make a selection, as shown in Eq. 1.

$$p_t(m_i) = \frac{exp\big(\mathcal{R}_t(m_t, C_t)/\tau\big)}{\sum_{m_j \in \mathcal{M}} exp\big(\mathcal{R}_t(m_t, C_t)/\tau\big)}. \tag{1}$$

[1]For brevity, $e_i := e_i^{(t)}$, *i.e.*, the candidate entity index for timestep t is not shown.

Here, τ is a temperature parameter that controls the degree of exploration. While Boltzmann exploration provides a simple yet efficient approach, the agent needs to sufficiently explore all models (arms) before making good decisions, which may be prohibitively expensive. Additionally, the need to tune the additional temperature parameter τ adds more computational overhead.

The family of UCB algorithms [2, 3] selects a bandit with the highest upper confidence bound. UCB-1 [2] maximizes

$$p_t(m_t) = \frac{\mathcal{R}_t(m_t, C_t)}{s_t} + \sqrt{\frac{2 \ln t}{s_t}}, \tag{2}$$

where s_t is the number of times the model m_t has been selected thus far and t is the time step (iteration).

In contrast, Thompson sampling [31] models $p_t(m_t)$ based on a Beta distribution derived from existing observations, *i.e.*,

$$p_t(m_t) = Beta\big(S_t(m_t) + 1, U_t(m_t) + 1\big), \tag{3}$$

where $S_t(m_t) = |\mathcal{P}_{m_t}^+|$ is the number of candidates from model m_t that were accepted in all previous time steps $1, \ldots, t-1$, and similarly $U_t(m_t) = |\mathcal{N}_{m_t}^-|$ is the number of candidates from model m_t that were rejected in all previous time steps $1, \ldots, t-1$. All previously described methods explore the models to some extent, but settle down once some sufficiently good models are found. On the other hand, the Exp3 algorithm continues searching for changes in the model rewards. The benefit to this approach is that sometimes set expansion models may perform differently with varying sizes of the set of positive and negative entities $\mathcal{P}^+$ and $\mathcal{N}^-$. However, this also has the risk of generating suboptimal results in general use cases. Exp3 introduces a new weight $w_t(m_t)$ for each arm (in this case, model candidate m_t) and uses these weights to select new candidates, *i.e.*,

$$p_t(m_t) = (1-\gamma)\frac{w_t(m_t)}{\sum_{m_j \in \mathcal{M}} w_j(m_j)} + \frac{\gamma}{M} \tag{4}$$

where $\gamma \in [0,1]$ is a factor controlling the probability of picking an action uniformly at random, and M is the total number of set expansion models. The weights are updated with

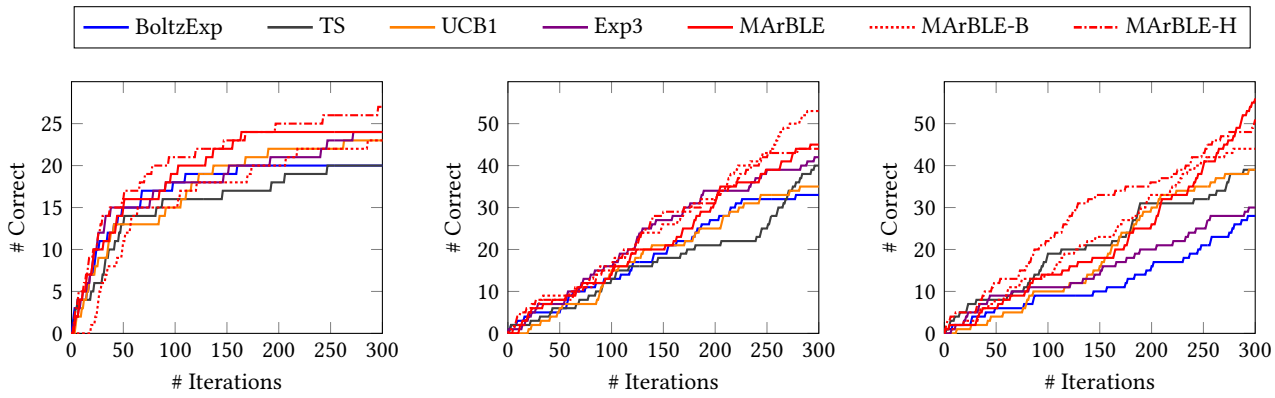
$$w_{t+1}(m_t) = \begin{cases} w_t(m_t) exp\left(\left(\frac{\mathcal{R}_t(m_t, C_t)}{p_t(m_t)}\gamma\right)/M\right) & \text{if } m_t \text{ is selected} \\ w_t(m_t) & \text{otherwise.} \end{cases} \tag{5}$$

In our experiments, we observe that each bandit algorithm can have suboptimal results depending on the domain and granularity of the semantics. In reality, it would be expensive to run experiments to discover the optimal algorithm among these bandit methods for a given set expansion task. Therefore, we propose MArBLE, a hierarchical multi-armed bandit method to adjust model selection based on task feedback. The proposed method leverages the intrinsic feedback component of human-in-the-loop set expansion tasks, without introducing any additional annotation burden or overhead.

3.3 MArBLE

Intuitively, MArBLE is a meta-learner that selects among a set of K bandit agents $\mathcal{A} = \{\mathcal{A}_1 \ldots \mathcal{A}_K\}$. At each time step t, MArBLE evaluates the set of bandit agents based on their rewards history[2] $R_t(\mathcal{A}_i) = \{\mathcal{R}_1(\mathcal{A}_i), \ldots, \mathcal{R}_{t-1}(\mathcal{A}_i)\}$. Specifically, at time step

[2]We simplify notation for reward for A_i at time step t, *i.e.*, $\mathcal{R}_t(\mathcal{A}_i) = \mathcal{R}_t^{(A_i)}(m_t, C_t)$



(a) RGB Color Names (b) ADE (c) FoodOn

Figure 3: Cumulative total number of accepted candidates.

t, MArBLE computes the posterior distribution $p(\mathcal{A}_i|R_t(\mathcal{A}_i))$ of selecting algorithm A_i as $p(\mathcal{A}_i|R_t(\mathcal{A}_i)) \propto p(\mathcal{A}_i)p(R_t(\mathcal{A}_i)|\mathcal{A}_i)$, where the marginal likelihood $p\left(R_t(\mathcal{A}_i)|\mathcal{A}_i\right)$ is given by

$$p\Big(R_t(\mathcal{A}_i)|\mathcal{A}_i\Big) = \frac{R_t(\mathcal{A}_i)}{\sum\limits_{A_j \in A} R_t(\mathcal{A}_j)}, \tag{6}$$

and the prior is assumed to be uniform, *i.e.*, $p(\mathcal{A}_i) = 1/|\mathcal{A}|$. Conversely, the Bayes Factor $\frac{p(R_t(\mathcal{A}_i)|\mathcal{A}_i)}{p(R_t(\mathcal{A}_j)|\mathcal{A}_j)}$ expresses the preference over two bandit agents $\mathcal{A}_i$ and $\mathcal{A}_j$. An overview of MArBLE is shown in Figure 2.

3.4 Cold-Start Problem

One limitation of MArBLE (and bandits in general) is that models may not have sufficient data for a new task to make a good guess. The lack of expert feedback at the initial stages makes all options equally possible. In other words, there is no domain knowledge that we can use a priori. Consequently, it is difficult to make inferences at the beginning. To alleviate this cold-start problem, we propose two boosting algorithm variants, explained below.

Frequency-based Boosting (MArBLE-B): Initially, many models may provide the same generic suggestions. One way to alleviate this is to collect a few initial suggestions from all models and present the most frequent ones to the subject matter expert. The benefit is that it is possible to provide the subject matter expert with a small set of entities and reward a large set of models based on the feedback at a very low cost. For frequency-voted boosting, we first acquire $k_1 \geq 0$ candidates from all models and select top-k_2 candidates based on majority voting, where $k_2 \leq k_1$. Then, we present the top-k_2 candidates to the subject-matter expert and receive feedback on which candidates are accepted. The accepted candidate entities are added to the initial positive set. In preliminary experiments, we observe that larger values of k_1 (>10) lead to more noisy entities. Additionally, we reward each model $m_i \in \mathcal{M}$ with $\mathcal{R}_t(m_t, C_t) = |\mathcal{P}^+_{t,m_i}|$, where $\mathcal{P}^+_{t,m_i}$ is the number of accepted entities at time step t from model m_i. In other words, the reward is proportional to the number of accepted entities found in the model's candidate list, thereby rewarding models that are more likely to be useful.

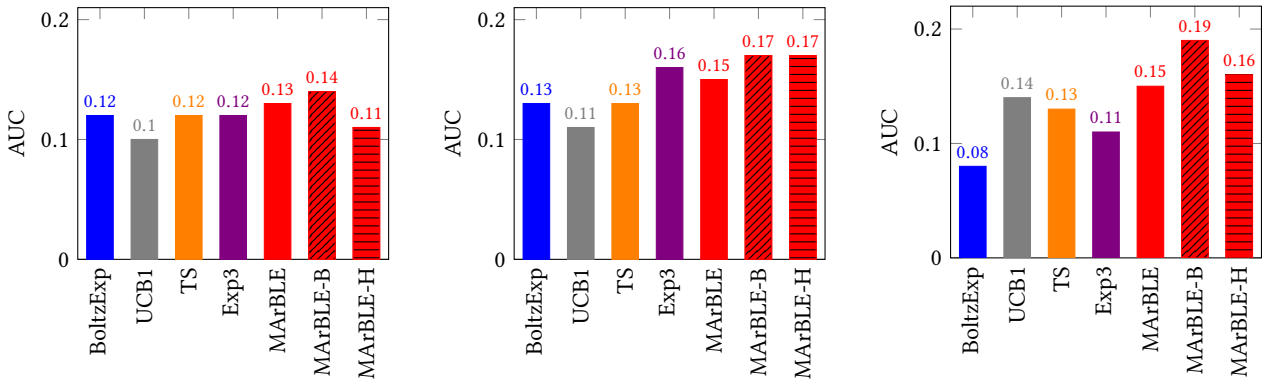
History-based Boosting (MArBLE-H): Although we show the subject matter expert $k = 1$ candidate entity at a time (top-k from each model), many of the other candidates lower in the list may also have been relevant. Some of these candidates may be provided to the subject-matter expert in future iterations, perhaps due to being selected from a different model in later rounds. It would make sense to also reward both the model that selects a candidate, as well as any other model that produces this candidate lower in their list in previous steps. In the history-based boosting setup, we take $k_1 \geq 0$ candidates in every iteration. We only present the top-k candidates to the subject-matter expert but keep the remaining $(k_1 - k)$ candidate entities in a hidden set. For each accepted entity at time step t, and in addition to rewarding the currently selected model m_t, we also reward any other models $m_{j,j\neq t} \in \mathcal{M}$ that had previously suggested the same candidate. We observe that MArBLE is robust on a range of k_2 values because for a model to receive a reward, the entity has to get accepted in a future iteration. This strategy allows for minimizing the number of candidates we pass to the subject matter expert for annotation (minimizing human effort) but maximizes the efficacy of the proposed approach by retrospectively rewarding models for lower-ranked candidates.

4 EXPERIMENTAL RESULTS

4.1 Experimental Setup

Models: We make use of methods available in the existing human-in-the-loop set expansion literature [8, 13]. We build 40 EnE [13] and 40 GLIMPSE [8] models, respectively, trained on 40 different datasets. In addition, we use a WordNet [11] model in our experiments. In total, our benchmark consists of 81 models acting as possible options to choose from in each experiment. As mentioned, each of the models takes as input a set of positive entities and a set of negative entities and returns a list of ranked candidate entities.

Datasets: We utilize three ground-truth lexicons from a diverse set of tasks as subject-matter expert proxies. For all three tasks, we randomly sample 10% of the lexicon as the initial entity seed set.
Adverse Drug Effects (ADE): The ADE corpus [15] consists of 3,341 adverse drug effects extracted from medical case reports.



(a) RGB Color Names (b) ADE (c) FoodOn

Figure 4: Area Under the Curve (AUC) after 300 iterations.

Some examples of entities in the ADE corpus are 'coronary spasm', 'aseptic meningitis', 'protamine allergy', and 'malignant lymphoma'.
RGB Color Names: We extract RGB color names from the X11 RGB file, which consists of 145 names of RGB colors. Some examples of entities are 'royal blue', 'brown', 'dark green', and 'cyan'.
FoodOn: We extract food names from the FoodOn ontology [9] that consists of 16, 801 names of various items consumed as food. Some examples of entities in FoodOn are 'shortbread cake small', 'cookie mix prepared from powder', and 'crowberry food product'.

Bandit Methods: We use a multi-armed bandit agent to select models at each iteration. We compare against the four different bandit algorithms described in Section 3.2: (a) Boltzmann Exploration (**BoltzExp**), (b) Upper Confidence Bound (**UCB1**), (c) Thompson Sampling (**TS**), and (d) **Exp3**.

4.2 Sequential Model Selection

To evaluate the correctness of the bandit algorithms in the set expansion tasks, we plot the number of correct entities extracted at each iteration, as shown in Figure 3. Although we see a similar pattern for all of the traditional bandit algorithms across the three tasks, their performance has significant differences. Moreover, there is no single best algorithm that performs best in all three tasks. For example, **Exp3** outperforms others in the first two tasks but performs significantly worse in the FoodOn task.

Additionally, we observe that the difference among the bandit algorithms becomes more prominent as the number of iterations grows. For example, for the RGB Color Names task, most of them converge pretty quickly, except for **BoltzExp**. However, for the FoodOn lexicon, there exists a large gap between the best and the worst algorithms. **UCB1** yields 39 accepted entities compared to the 28 accepted entities extracted using **BoltzExp**. In terms of relative gain, after 300 iterations, **UCB1** generates 39.29% more accepted entities compared to **BoltzExp** on FoodOn. Similarly, **Exp3** generates 21.21% more entities compared to **BoltzExp** on Adverse Drug Effects. In terms of AUC, this represents a 75.49% and 27.27% increase on FoodOn and Adverse Drug Effects, respectively. This clearly shows the significance of selecting an appropriate bandit algorithm. Similar findings are observable in Figure 4.

However, selecting an appropriate method for each task can be non-trivial and expensive both in terms of computational resources and in terms of subject-matter expert effort. MArBLE facilitates automatic selection with solid performance vs. computational overhead trade-offs. Experimental results show that MArBLE performs well across all three lexicons. In addition, MArBLE outperforms **UCB1** by a large margin in the FoodOn task, achieving 56.67% more accepted entities. Similarly, MArBLE outperforms **Exp3** in the Adverse Drug Effects task, achieving 7.14% more accepted entities. Likewise, MArBLE retrieves the same number of entities as **Exp3** in the RGB Color Names task. Additionally, MArBLE-B surpasses the performance of all the traditional bandit algorithms across all three tasks. We further observe that MArBLE-B outperforms MArBLE and all the baselines in terms of AUC.

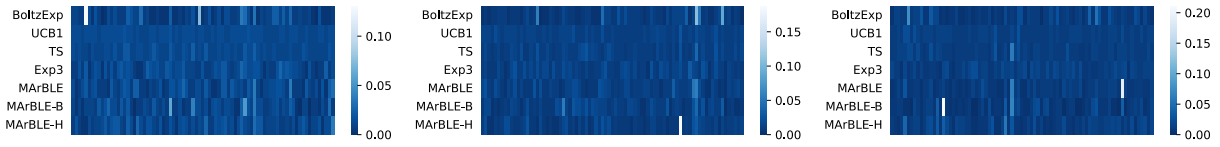
4.3 Qualititative Analysis

To understand how model selection differs among each bandit, we qualitatively analyze the percentage of times each algorithm selects different models in Figure 5 On each row, a brighter color means that the model on the corresponding column was selected a higher number of times. We observe that **BoltzExp** has a large number of models that were selected very often. This is expected as **BoltzExp** is known for more exploitation and less exploration, which has, in turn, resulted in suboptimal results in terms of accepted entities. MArBLE variants find a few good model candidates but also sufficiently explore the model space.

We then take a closer look at some of the most frequently selected models. In Table 1, we observe that some of the top model choices for the FoodOn task, across all bandits, are models trained on social media and food-related datasets, for example, `Social Media Posts on NYC` and `PizzaWP`. There are also some models trained on general datasets, for example, `UKWAC`, a British English corpus from the .uk domain. On the other hand, a popular choice for the ADE task is models trained on medical-related datasets, *i.e.*, `MIMIC II`, `MIMIC III`, `DynaMed`, `COVID19`, `NEJM` (New England

Table 1: Top-3 most frequently selected models. Columns #selected denote the number of times an algorithm selected a model, and #accepted is the number of candidate entities accepted by the subject-matter expert for the corresponding model.

	RGB Color Names			ADE			FoodON		
	model	# selected	# accepted	model	# selected	# accepted	model	# selected	# accepted
BoltzExp	EnE: SocMedNycLoc	39	3	GLIMPSE: MIMIC II	32	7	EnE: SocMedNyc	23	5
	EnE: Wiki 2018	22	2	GLIMPSE: OsteoReports	26	2	GLIMPSE: Finance Contracts I	16	1
	GLIMPSE: Astronomy	11	2	EnE: ArticlesCOVID	17	5	GLIMPSE: Dynamed	12	1
UCB1	EnE: Tweets2016	5	2	GLIMPSE: Dynamed	8	5	EnE: Wiki 2018	8	5
	EnE: Chemical Journals	5	2	EnE: NEJM	6	3	EnE: SocMedNyc	7	4
	GLIMPSE: DiabetesMedTexts	5	2	EnE: SocMedNycLoc	6	3	EnE: SocMedGatherings	7	4
TS	GLIMPSE: Chemical Journals	7	2	GLIMPSE: MIMIC II	12	6	EnE: ukWaC	18	11
	EnE: ukWaC	7	2	EnE: PubMed	8	3	EnE: PubMed	9	4
	EnE: MIMIC III	6	1	EnE: ArticlesCOVID	7	2	EnE: Astronomy	7	3
Exp3	GLIMPSE: Astronomy	8	1	GLIMPSE: HealthImagingReports	9	0	EnE: MIMIC II	7	0
	GLIMPSE: HealthImagingReports	8	0	EnE: Product Catalog	7	0	EnE: MIMIC III	7	3
	GLIMPSE: Product Catalog	7	0	EnE: Airline Help Tickets	7	0	GLIMPSE: Sales Contracts	6	0
MArBLE	GLIMPSE: Medical Transcription	11	1	GLIMPSE: MIMIC II	17	8	GLIMPSE: PizzaWP	57	25
	GLIMPSE: Chemical Journals	11	2	GLIMPSE: MIMIC III	15	6	EnE: ukWaC	19	10
	WordNet	9	1	GLIMPSE: Radiographic Reports	10	2	EnE: SocMedNycLoc	9	5
MArBLE-B	EnE: Tweets2016	18	2	EnE: MIMIC III	16	4	EnE: Chemical Journals	63	20
	GLIMPSE: Astronomy II	16	0	GLIMPSE: MIMIC II	12	4	EnE: ukWaC	20	10
	EnE: ukWaC	12	1	GLIMPSE: OsteoReports	9	2	EnE: DiabetesMedTexts	11	4
MArBLE-H	WordNet	11	1	GLIMPSE: DynaMed	56	18	EnE: Tweets from 2020	17	7
	GLIMPSE: Chemical Journals	9	2	GLIMPSE: MIMIC II	9	4	EnE: SocMedNycLoc	13	6
	GLIMPSE: Sales Contracts	8	2	EnE: DiabetesMedTexts	8	2	EnE: Chemical Journals	11	5



(a) RGB Color Names (b) ADE (c) FoodON

Figure 5: Model selection distributions for each algorithm. Number of times each selection algorithm selects different models: each column corresponds to one of the 81 set expansion models, and each row corresponds to the selection algorithms.

Journal of Medicine), *etc.* This further demonstrates that the proposed method can successfully discover useful models for the task at hand. Another interesting observation is that most of the top choices of **Exp3** yield very few, if any, accepted entities. This is a known phenomenon for **Exp3**, where the algorithm keeps exploring in the hopes of a change in the reward distribution, resulting in suboptimal results in traditional bandit settings [27].

5 SYSTEM DEPLOYMENT AND USE

MArBLE and its early prototypes have been deployed since 2020, with a patent filed and pending. In our experiments, the MArBLE-B variant seems to better deal with the cold-start problem and yields a significantly higher AUC. Therefore, we have MArBLE-B deployed in our production systems. These systems have been used to perform set expansions for over two dozen clients, in fields ranging from fashion to product concerns to organized retail crime. In all these cases we have been able to expand lexicons to achieve better recall by working into the long tail of the concepts of interest. The MArBLE approach lets subject matter experts who are not computer scientists provide invaluable training data to the underlying AI systems. In real-world applications, we have observed that human experts are more efficient in adjudicating suggested lexicon entries compared to generating new samples without any prompts. Adjudicating a single term takes only seconds, while generating new samples becomes increasingly time-consuming, with several hours required for a few hundred lexicon entries. In our deployed system, subject matter experts spend an average of approximately ~2 seconds per adjudication. Adjudicating a set of 300 samples would take around 10 minutes. These observations highlight the significant time savings achieved by leveraging expert adjudication over generating new entries independently.

6 CONCLUSION

Selecting suitable models for a particular task is often non-trivial as such selection requires significant a priori knowledge about the method, the model architecture, and the training data available. In this work, we explore human-in-the-loop set expansion and utilize bandit strategies to automatically select models based on a subject-matter expert's feedback. We propose a hierarchical multi-armed bandit approach, termed MArBLE, as a meta-learner that selects among the set of bandit methods, and two boosting methods to improve performance in cold-start scenarios. MArBLE alleviates the need to select appropriate exploration mechanisms. Future work can extend to handle noisy oracles and combine models performing diverse but interconnected tasks.

7 ACKNOWLEDGEMENTS

This material is based upon work supported by the National Science Foundation under Grant No. 2208864.

REFERENCES

[1] Alfredo Alba, Anni Coden, Anna Lisa Gentile, Daniel Gruhl, Petar Ristoski, and Steve Welch. 2017. Multi-lingual concept extraction with linked data and human-in-the-loop. In *K-CAP*. 1–8.

[2] Peter Auer. 2002. Using confidence bounds for exploitation-exploration trade-offs. *JMLR* Nov (2002), 397–422.

[3] Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. 2002. Finite-time analysis of the multiarmed bandit problem. *Machine learning* 47, 2 (2002), 235–256.

[4] Samuel Budd, Emma C Robinson, and Bernhard Kainz. 2021. A survey on active learning and human-in-the-loop deep learning for medical image analysis. *Medical Image Analysis* 71 (2021), 102062.

[5] Deepayan Chakrabarti, Ravi Kumar, Filip Radlinski, and Eli Upfal. 2008. Mortal multi-armed bandits. In *NeurIPS*.

[6] Jun Chen, Yueguo Chen, Xiangling Zhang, Xiaoyong Du, Ke Wang, and Ji-Rong Wen. 2018. Entity set expansion with semantic features of knowledge graphs. *Journal of Web Semantics* 52 (2018), 33–44.

[7] Wei Chen, Yajun Wang, and Yang Yuan. 2013. Combinatorial multi-armed bandit: General framework and applications. In *ICML*. PMLR, 151–159.

[8] Anni Coden, Daniel Gruhl, Neal Lewis, Michael Tanenblatt, and Joe Terdiman. 2012. Spot the drug! an unsupervised pattern matching method to extract drug names from very large clinical corpora. In *IEEE Second International Conference on Healthcare Informatics, Imaging and Systems Biology*. IEEE, 33–39.

[9] Damion M Dooley, Emma J Griffiths, Gurinder S Gosal, Pier L Buttigieg, Robert Hoehndorf, Matthew C Lange, Lynn M Schriml, Fiona SL Brinkman, and William WL Hsiao. 2018. FoodOn: a harmonized food ontology to increase global food traceability, quality control and data integration. *NPJ Science of Food* 2, 1 (2018), 1–10.

[10] Crícia Z Felício, Klérisson VR Paixão, Celia AZ Barcelos, and Philippe Preux. 2017. A multi-armed bandit model selection for cold-start user recommendation. In *UMAP*. 32–40.

[11] Christiane Fellbaum and K Brown. 2005. Encyclopedia of language and linguistics.

[12] Alexis Fouilloy, Cyril Voyant, Gilles Notton, Fabrice Motte, Christophe Paoli, Marie-Laure Nivet, Emmanuel Guillot, and Jean-Laurent Duchaud. 2018. Solar irradiation prediction with machine learning: Forecasting models selection method depending on weather variability. *Energy* 165 (2018), 620–629.

[13] Anna Lisa Gentile, Daniel Gruhl, Petar Ristoski, and Steve Welch. 2019. Explore and exploit. Dictionary expansion with human-in-the-loop. In *ESWC*. Springer, 131–145.

[14] Zoubin Ghahramani and Katherine A Heller. 2005. Bayesian Sets. In *NeurIPS*. 435–442.

[15] Harsha Gurulingappa, Abdul Mateen Rajput, Angus Roberts, Juliane Fluck, Martin Hofmann-Apitius, and Luca Toldo. 2012. Development of a benchmark corpus to support the automatic extraction of drug-related adverse effects from medical case reports. *Journal of Biomedical Informatics* 45, 5 (2012), 885–892.

[16] Ryosuke Kohita, Issei Yoshida, Hiroshi Kanayama, and Tetsuya Nasukawa. 2020. Interactive construction of user-centric dictionary for text analytics. In *ACL*. 789–799.

[17] Yen-Hsien Lee, Tsang-Hsiang Cheng, Ci-Wei Lan, Chih-Ping Wei, and Paul Jen-Hwa Hu. 2009. Overcoming small-size training set problem in content-based recommendation: a collaboration-based training set expansion approach. In *ICEC*. 99–106.

[18] Yinghui Li, Yangning Li, Yuxin He, Tianyu Yu, Ying Shen, and Hai-Tao Zheng. 2022. Contrastive learning with hard negative entities for entity set expansion. In *SIGIR*. 1077–1086.

[19] Blerina Lika, Kostas Kolomvatsos, and Stathes Hadjiefthymiades. 2014. Facing the cold start problem in recommender systems. *Expert Systems with Applications* 41, 4 (2014), 2065–2073.

[20] Jingjing Liu, Xiao Li, Alex Acero, and Ye-Yi Wang. 2011. Lexicon modeling for query understanding. In *ICASSP*. IEEE, 5604–5607.

[21] Ismini Lourentzou, Daniel Gruhl, Alfredo Alba, Anna Lisa Gentile, Petar Ristoski, Chad Deluca, Steven R Welch, and Chengxiang Zhai. 2021. AdaReNet: Adaptive Reweighted Semi-supervised Active Learning to Accelerate Label Acquisition. In *The 14th PErvasive Technologies Related to Assistive Environments Conference*. 431–438.

[22] Ismini Lourentzou, Daniel Gruhl, and Steve Welch. 2018. Exploring the efficiency of batch active learning for human-in-the-loop relation extraction. In *Companion Proceedings of the The Web Conference 2018*. 1131–1138.

[23] Patrick Pantel, Eric Crestan, Arkady Borkovsky, Ana-Maria Popescu, and Vishnu Vyas. 2009. Web-scale distributional similarity and entity set expansion. In *EMNLP*. 938–947.

[24] Pushpendre Rastogi, Adam Poliak, Vince Lyzinski, and Benjamin Van Durme. 2019. Neural variational entity set expansion for automatically populated knowledge graphs. *Information Retrieval Journal* 22, 232–255.

[25] Kugatsu Sadamitsu, Kuniko Saito, Kenji Imamura, and Yoshihiro Matsuo. 2012. Entity set expansion using interactive topic information. In *PACLIC*. 108–116.

[26] Luis Sarmento, Valentin Jijkuon, Maarten De Rijke, and Eugenio Oliveira. 2007. "More like these" growing entity classes from seeds. In *CIKM*. 959–962.

[27] Yevgeny Seldin, Csaba Szepesvári, Peter Auer, and Yasin Abbasi-Yadkori. 2013. Evaluation and analysis of the performance of the EXP3 algorithm in stochastic environments. In *EWRL*. PMLR, 103–116.

[28] Jiaming Shen, Wenda Qiu, Jingbo Shang, Michelle Vanni, Xiang Ren, and Jiawei Han. 2020. SynSetExpan: An iterative framework for joint entity set expansion and synonym discovery. In *EMNLP*. Association for Computational Linguistics (ACL), 8292–8307.

[29] Chuan Shi, Jiayu Ding, Xiaohuan Cao, Linmei Hu, Bin Wu, and Xiaoli Li. 2021. Entity set expansion in knowledge graph: a heterogeneous information network perspective. *Frontiers of Computer Science* 15, 1 (2021), 1–12.

[30] Richard S Sutton and Andrew G Barto. 2018. *Reinforcement learning: An Introduction*. MIT press.

[31] William R Thompson. 1933. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika* 25, 3-4 (1933), 285–294.

[32] Paola Velardi, Stefano Faralli, and Roberto Navigli. 2013. Ontolearn reloaded: A graph-based algorithm for taxonomy induction. *Computational Linguistics* 39, 3 (2013), 665–707.

[33] Richard C Wang and William W Cohen. 2007. Language-independent set expansion of named entities using the web. In *ICDM*. IEEE, 342–350.

[34] Richard C Wang and William W Cohen. 2008. Iterative set expansion of named entities using the web. In *ICDM*. IEEE, 1091–1096.

[35] Richard C Wang, Nico Schlaefer, William Cohen, and Eric Nyberg. 2008. Automatic set expansion for list question answering. In *EMNLP*. 947–954.

[36] Ye-Yi Wang, Raphael Hoffmann, Xiao Li, and Jakub Szymanski. 2009. Semi-supervised learning of semantic classes for query understanding: from the web and for the web. In *CIKM*. 37–46.

[37] Xingjiao Wu, Luwei Xiao, Yixuan Sun, Junhang Zhang, Tianlong Ma, and Liang He. 2022. A survey of human-in-the-loop for machine learning. *Future Generation Computer Systems* (2022).

[38] Doris Xin, Litian Ma, Jialin Liu, Stephen Macke, Shuchen Song, and Aditya Parameswaran. 2018. HELIX: Accelerating Human-in-the-loop Machine Learning. In *VLDB Endowment*, Vol. 11.

[39] Lingyong Yan, Xianpei Han, Ben He, and Le Sun. 2020. End-to-end bootstrapping neural network for entity set expansion. In *AAAI*. 9402–9409.

[40] Lingyong Yan, Xianpei Han, Ben He, and Le Sun. 2020. Global bootstrapping neural network for entity set expansion. In *EMNLP*. 3705–3714.

[41] Lingyong Yan, Xianpei Han, Le Sun, and Ben He. 2019. Learning to bootstrap for entity set expansion. In *EMNLP-IJCNLP*. 292–301.

[42] Lei Zhang and Bing Liu. 2011. Entity set expansion in opinion documents. In *Proceedings of the 22nd ACM Conference on Hypertext and Hypermedia*. 281–290.

[43] Xiangling Zhang, Yueguo Chen, Jun Chen, Xiaoyong Du, Ke Wang, and Ji-Rong Wen. 2017. Entity set expansion via knowledge graphs. In *SIGIR*. 1101–1104.

[44] Yunyi Zhang, Jiaming Shen, Jingbo Shang, and Jiawei Han. 2020. Empower Entity Set Expansion via Language Model Probing. In *ACL*. 8151–8160.

[45] Yuyan Zheng, Chuan Shi, Xiaohuan Cao, Xiaoli Li, and Bin Wu. 2017. Entity set expansion with meta path in knowledge graph. In *PAKDD*. Springer, 317–329.

[46] ZAIM Zulkifly, KYAIRUL AZMI Baharin, and CHIN KIM Gan. 2021. Improved Machine Learning Model Selection Techniques for Solar Energy Forecasting Applications. *IJRER* 11, 1 (2021), 308–319.