SciMine: An Efficient Systematic Prioritization Model Based on Richer Semantic Information

Fang Guo*
School of Engineering, Westlake University
Hangzhou, China
guofang@westlake.edu.cn

Yun Luo*
School of Engineering, Westlake University
Hangzhou, China
luoyun@westlake.edu.cn

Linyi Yang
School of Engineering, Westlake University
Hangzhou, China
yanglinyi@westlake.edu.cn

Yue Zhang
School of Engineering, Westlake University
Hangzhou, China
zhangyue@westlake.edu.cn

ABSTRACT

Systematic review is a crucial method that has been widely used by scholars from different research domains. However, screening for relevant scientific literature from paper candidates remains an extremely time-consuming process so the task of screening prioritization has been established to reduce the human workload. Various methods under the human-in-the-loop fashion are proposed to solve this task by using lexical features. These methods, even though achieving better performance than more sophisticated feature-based models such as BERT, omit rich and essential semantic information, therefore suffered from feature bias. In this study, we propose a novel framework SciMine to accelerate this screening process by capturing semantic feature representations from both background and the corpus. In particular, based on contextual representation learned from the pre-trained language models, our approach utilizes an autoencoder-based classifier and a feature-dependent classification module to extract general document-level and phrase-level information. Then a ranking ensemble strategy is used to combine these two complementary pieces of information. Experiments on five real-world datasets demonstrate that SciMine achieves state-of-the-art performance and comprehensive analysis further shows the efficacy of SciMine to solve feature bias.

CCS CONCEPTS

• **Information systems** → **Users and interactive retrieval**; • **Theory of computation** → **Active learning**.

KEYWORDS

Screening Prioritization, Systematic Review, Meta Analysis, Human-in-the-loop, Active Learning, Text Classification

ACM Reference Format:
Fang Guo*, Yun Luo*, Linyi Yang, and Yue Zhang. 2023. SciMine: An Efficient Systematic Prioritization Model Based on Richer Semantic Information. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '23), July 23–27, 2023, Taipei, Taiwan.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3539618.3591764

*Both authors contributed equally to the paper

SIGIR '23, July 23–27, 2023, Taipei, Taiwan

ACM ISBN 978-1-4503-9408-6/23/07.
https://doi.org/10.1145/3539618.3591764

1 INTRODUCTION

Systematic review aims to use systematic and explicit methods to collect, identify and critically appraise relevant studies about one research theme [12, 54]. It is an important research method that scholars from different domains like Medicine, Agriculture, and Biology have widely used. After querying databases of science literature, scholars need to screen the retrieved unordered set of paper candidates to ensure the comprehensiveness and correctness of the systematic review. Since the screening process can be highly expensive and time-consuming, the task of screening prioritization is established to reduce the human workload. Formally, screening prioritization refers to the task of searching for relevant documents given an unordered set of paper candidates. Various automatic systems have been developed to learn from user needs based on their screening record and return the relevant papers to them.

Recent studies have demonstrated the effectiveness of building an active learner to solve this task [10, 13, 44, 54, 56, 67]. As shown in Figure 1, when the active learner performs each human-in-the-loop iteration, a classifier is trained on the set of labeled documents and predicts the set of unlabeled candidates to find the one that is most likely to be relevant. The user then screens this document and backfed it to the learner for incremental training. Most applications propose to build the classifier based on lexical features [13, 44, 54, 56]. Current state-of-the-art model AsReview [54], which uses TF-IDF as feature extraction with a Naive Bayers Classifier can outperform models that use more sophisticated feature extractions like Doc2vec and BERT. Due to the sparsity of scientific phrases in the corpus, however, these models may deviate to focus on spurious patterns [19, 30, 32, 53]. Intuitively, models should not neglect the semantic information from both corpus and background knowledge.

Making use of the richer semantic information, though appealing, poses its own challenges. The first challenge is how to infer background knowledge across different scientific domains. Pre-trained language model seems a good fit but recent studies [58, 64] also show that the PLM-based neural rankers can not guarantee better performance over lexical-based methods. Second, the classifier may suffer from feature bias. This is due to the number of relevant documents being far fewer than irrelevant documents so

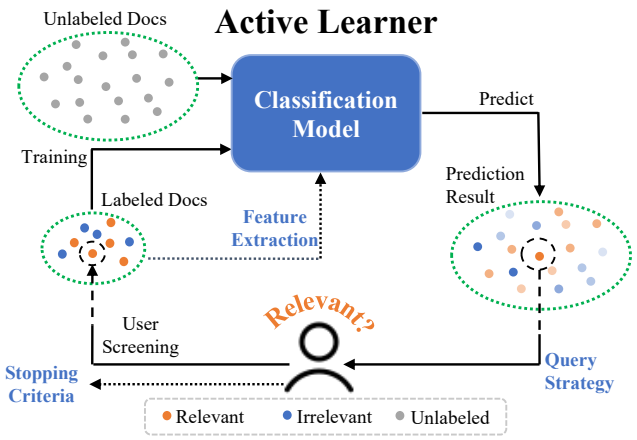

Figure 1: Pipeline of the human-in-the-loop iteration performed by SciMine.

that the classifier may learn from frequent but theme-irrelevant phrases. We further analyze this phenomenon in Section 4.2. The third challenge comes from the difficulty of capturing the minimal difference between relevant and irrelevant documents. For example, in a study whose research theme is nudging healthcare professionals, the representation of "Just-in-time evidence-based e-mail reminders in home health care: impact onnurse practices" and "Just-in-time evidence-based e-mail reminders in home health care: impact on patient outcomes" is very close in the latent space since they have many phrases in common. However, the domain expert can tell only the first one is relevant because "onnurse" is a phrase more related to "healthcare professionals" than "patient".

To address the above challenges, we consider a framework that mimics the domain expert behavior, by categorizing documents using scientific background knowledge and semantic relevant features. We name the framework SciMine, which consists of three main components, as shown in Figure 2. First, in order to obtain domain knowledge, we adopt SPECTER [11], a Transformer language model pre-trained on the citation network of scientific literature, to obtain document embeddings and a pre-trained masked language model (MLM) to learn phrase embeddings. Second, to cope with feature bias, we adopt a variational autoencoder model to rank the candidate documents while preserving semantic information from pre-trained representations. Third, to further integrate phrase-based rationale, we use a community detection algorithm to find phrase-level features and train a classifier on these features to provide a second ranking of the candidates. Finally, two rankings are merged using an ensemble approach. We adopt the standard query strategy and experimental settings of previous work [16, 54]. Experiments on five standard benchmarks show that SciMine outperforms existing methods significantly, achieving the best-reported results in the literature. In addition, for a detailed understanding of the underlying mechanisms, we conducted a human study with ecological experts, resulting in a new dataset **AgriDiv**, which consists of 1,505 documents, and 129 of them are relevant to the research theme: "investigating the agriculture diversification in rice production".

In summary, we build SciMine, a human-in-the-loop framework for efficient screening prioritization. To our knowledge, we are the first to show the efficacy of adopting contextual representation from the pre-trained language model in the task. SciMine can save more than 10% workload than the current SOTA. Furthermore, to better understand intrinsic user screening habits in the systematic review, we work with ecological scientists to create a novel dataset, **AgriDiv**, which includes research papers in the ecological domain. We perform a user study on this dataset and gain some valuable observations for future work. We open-source the codebase and the dataset[1].

2 RELATED WORK

Screening Priorization refers to the task of searching for relevant documents among an unordered set of paper candidates. A series of machine learning-evolved applications [1, 13, 16, 18, 44, 46, 49, 54, 56, 59, 67, 72] have been proposed for screening priorization. These models can be mainly categorized into "One-off" learning [58] and iterative learning. For "One-off" learning, seed information like the user's search query [1, 49, 62], the theme of the research [50], or a set of prior knowledge [25, 57, 59] is used to directly rank the document candidates. And for human-in-the-loop iterative learning, the model can iteratively accumulate user feedback, hence showing more efficiency in screening, and providing an intuitive user experience. Models under this fashion are varied in model input, query strategy, retraining strategy, and stopping criteria. For example, FASTREAD [67] utilizes uncertainty-based sampling to query documents for labeling and it retrains every ten iterations. Rayyan [44] takes user-provided words and citations as input and stops when the model can no longer be improved. CAL [13, 14, 18] and ASReview [54] both take a set of labeled documents as input, but the former one designs a "knee" method to automatically stop the iterations while the later one lets the user decide when to stop. One common point of these above human-in-the-loop learning works is they all classify over lexical features like TF-IDF. Another recent work [64] tests fine-tuning the BERT model in every iteration and concludes that this method underperforms the lexical-based method when the corpus has very different textual characteristics. Thus, while studies [16, 20] have demonstrated the power of lexical features via comprehensive experiments, the power of representation generated from pre-trained language models for this task remains unstudied. In this work, we not only analyze why the advanced contextual representation can beat lexical features, but also propose a model that can utilize document-level and phrase-level information.

Active Learning is a type of machine learning that allows the model to choose the training samples it would like to learn from. It has been widely used in text classification [3, 15, 45, 51, 66, 69] by concentrating the human annotating effort on the most informative data points that can boost model performance significantly [30, 35]. The problem setting for AL in text classification is to let the model query and train on a certain number of samples from the training set in each iteration, then test the performance of the model on a different test set. Though our model is one kind of active learning model, there are mainly two differences between

[1] https://github.com/fangguo1/SciMine

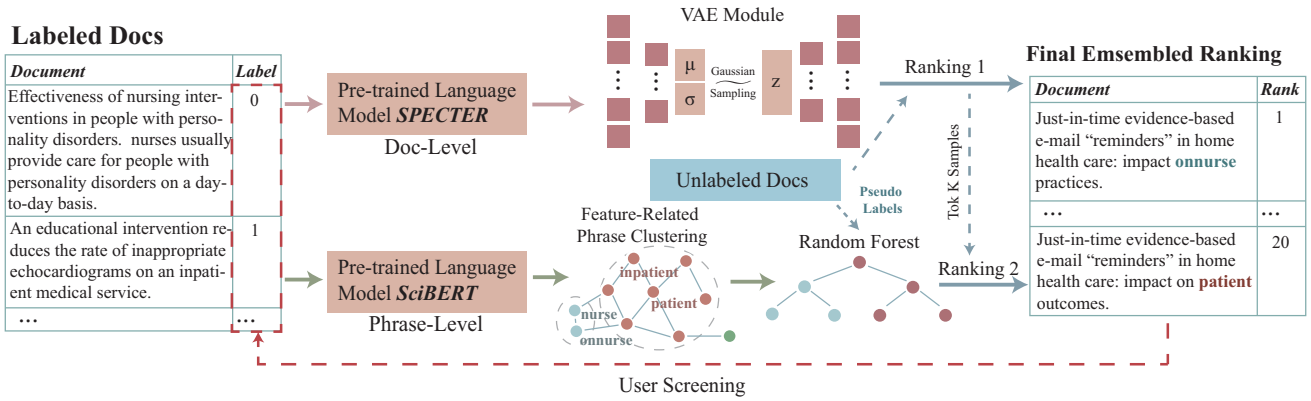

Figure 2: The framework of our proposed SciMine. It has three major components, a VAE-based document classification module, a phrase-level feature classification module, and a ranking ensemble module.

our work and theirs. First, previous AL methods in text classification query much more documents than our model in each iteration (**a few hundred vs one**). Second, the goal of previous work is to enhance the performance on the test set while we aim to minimize the number of human-in-the-loop iterations. In our study, we also test how the uncertainty-based contrastive query strategy from active learning influences our model.

Scientific Literature Learning aims to learn the representation of scientific documents to facilitate subtasks like text classification and relevant document search [2, 34, 43]. Some work on paper recommendation [6, 20, 48] utilizes additional author or citation information to recommend papers while we study how to recommend purely based on the paper's title and abstract. By using Transformers [55], these models usually pretrain on large domain-specific data [21, 26, 47] or data from multiple domains [5, 11, 32]. Besides learning from the text of the paper, they often employ citation features to capture the inter-document relations [11, 42]. Although the Transformer-based model dominates in a lot of scientific literature-related tasks, previous attempts to adapt PLM in screening prioritization [58] always fall short with classic lexical-based methods. Therefore, how the representation learned using PLM can boost the screening process remains an unsolved problem. In this work, we utilize PLM to generate both document embedding and phrase embedding and demonstrate that this contextualized information can outperform lexical features.

Text Classification with less training data has been studied for a long time and several lines of work have been proposed. Semi-supervised classification models [9, 40, 60, 63] generate augmented instances via creating real text segments or the hidden states of the model. Zero-shot text classification [28, 70] generalizes the knowledge learned from seen classes and transfers it to unseen ones. Even though this line of work requires less training data when compared to traditional classification models, they still ask for more human annotations than ours to kick off. Weakly-supervised classification [36–39, 61] tries to categorize documents based on the word-level description by using seed information like category-related words. However, the practical user need of our task is the model should train fast in each iteration while it usually takes a long time to train a weakly-supervised model. In addition, these models rely on the correlation between words and topics. But in our task, due to the complexity of the scientific literature, it is difficult to define a topic. In SciMine, we design a phrase-level feature classification module to help the document-level classifier by detecting important phrase-level features from the corpus.

3 TASK DEFINITION

Formally, a systematic review corpus ($\mathcal{D}$) is about one particular research theme and is collected by scholars querying databases of scientific literature. A candidate document $\mathbf{d}$ in this corpus is either relevant to the scholars' research theme ($\mathbf{d} \in \mathcal{R}$) or irrelevant ($\mathbf{d} \in \mathbf{I}$). To facilitate scholars finding relevant documents for their research, an active learner learns and finds relevant documents iteratively. As shown in figure 1, a complete human-in-the-loop iteration $\mathbf{t}$ contains the following steps: (1) a classification model is (re)trained on a set of user-labeled document $\mathcal{D}_l$ and predicts on the remaining unlabeled document set $\mathcal{D}_u$, (2) The active learner ranks documents from $\mathcal{D}_u$ and returns the top-ranking document to the user and, (3) the user reads this documents and decides whether it is relevant, where the user decision is used as a label that is back-fed to the learner, which then moves this labeled document from $\mathcal{D}_u$ to $\mathcal{D}_l$ for incremental training. In real use cases, this iteration repeats until the user feels there are few relevant documents in $\mathcal{D}_u$ and decides to stop. In our experiments, we also follow existing work [16, 54] and set a target percentage $\mathbf{p}$ of the relevant documents and study how to minimize the total iterations $\mathcal{T}$ needed to reach this target.

In the task of efficient screening prioritization, given (1) a systematic review corpus $\mathcal{D}$, where each document $\mathbf{d} \in \mathcal{D}$ is the concatenation of a research paper's title and abstract, (2) a seed set of user-labeled documents D_l. The label is binary, indicating whether a document is relevant (1) or not (0), and (3) a remaining set of unlabeled documents D_u. We aim to find the target percentage $\mathbf{p}$ of relevant documents D_l while minimizing the total iterations $\mathcal{T}$.

4 METHOD

We propose a novel active learner SciMine to address the problem of screening prioritization. As shown in figure 2, it has four steps: representation learning, VAE-based document level classification, phrase-level feature classification, and ranking ensemble. In this section, we introduce our proposed method by first introducing

how we learn both document-level and phrase-level representation in Section 4.1, then describe the two modules of our active learner in Sections 4.2 and 4.3 and the ranking ensemble module in Section 4.4.

4.1 Representation Learning

We first learn both document embedding and phrase embedding using pre-trained language models. For document-level representation, we apply a Transformer language model SPECTER [11] to generate document embeddings. SPECTER is pre-trained on a large scientific literature corpus and captures the relatedness between documents via a structure called "citation graph". In our case, we feed the concatenation of a paper's title and abstract into SPECTER and take the final representation of the [CLS] token as the embedding of the paper:

$$v = SPECTER([CLS]+Title+[SEP]+Abs+[SEP])_{[CLS]}. \quad (1)$$

For phrase-level representation, we first obtain quality phrases in the corpus by using a phrase mining tool called Autophrase [29, 52], then learn their MLM-based embeddings. For each phrase, we get its MLM-based embedding to capture both content and context features simultaneously. Suppose that a phrase p appears N_p times in the corpus. Then, for each of its mention p^l, $l \in \{1, 2, \ldots, N_p\}$, we obtain its *content feature* $\mathbf{x}_p^l$ by feeding the original sentence into a pre-trained MLM and taking the average of the generated embedding vectors corresponding to the tokens of p. To get the *context feature* $\mathbf{y}_p^l$ of this mention, we first replace the entire phrase p with a single [MASK] token, feed the new sentence into the same language model, and then use the embedding vector of this [MASK] token as the context feature. Finally, to get the phrase embedding that captures both content and context features, we concatenate two feature vectors for each mention and take the average of the resulting mention vectors:

$$\mathbf{e}_p = \frac{1}{N_p} \sum_{l=1}^{N_p} [\mathbf{x}_p^l; \mathbf{y}_p^l]. \quad (2)$$

Then we introduce how we utilize the representation information in SciMine.

4.2 AE-based Document-level Classification

Our neural model can be mainly divided into a feature extractor and a classifier, and we use cross-entropy loss to fit the model. We first demonstrate the feature bias by using the cross-entropy loss below.

Proposition 1: Minimal cross-entropy (CE) loss does not imply that all possible features of a task can be learned in the feature extractor.

Let us assume that all features of a class are learned with minimal CE loss. We use a toy example in **Virus** dataset, whose research theme is related to common livestock, to demonstrate the incorrectness. Assume that in the training data, all the positive documents have the word "piglet" and negative ones have the word "dog". We further assume that the feature extractor learns only two significant features in the feature vector by chance, one for the word "piglet" and one for the word "dog". When a study with the word "piglet" is presented to the feature extractor, the first feature has the value of 1 and all the other features have the value of 0, and for a sentence with the word "dog", all the features have the value of 0. Under this feature setting, we can easily achieve a cross-entropy value of 0 by adjusting weight values. This shows that a model can rely on incomplete and spurious patterns to fit the CE loss during training, but can fail to generalize during testing.

Proposition 2: Feature bias can easily occur in the task of iterative screening prioritization.

Among candidate documents for screening, the proportion of relevant documents is much smaller when compared with irrelevant documents. This leads to the feature extractor learning a subset of the features due to the limited number of relevant documents. Following the example mentioned above, the limited positive data may contain frequent but theme-irrelevant phrases such as "stool" and "fecal", which makes the classifier learn from these phrases while overlooking the features of less frequent but theme-relevant phrases like "poultry". This leads to feature bias. An empirical t-SNE visualization of features from an MLP feature extractor for classification is shown in Appendix A.1.

The above issues suggest that preserving feature information is crucial for the task. To this end, autoencoder can serve as a tool, which enforces that the same input can be reconstructed from a representation [8, 31, 33]. Here we adopt one of such methods – variational autoencoder (VAE) as our classification model, which uses the reconstruction loss to preserve the original semantic information and adds Gaussian noise to generate meaningful-semantic representations for isotropy [27, 68, 71].

Training. Suppose that we have the data v samples from the distribution parameterized by the ground truth generative factors z, VAE aims to maximize the probability of the v on average over all the possible samples from the latent factors, corresponding to:

$$max_{\Phi,\theta} \mathbb{E}_{q_\Phi}[log\ p_\theta(v|z)] \quad (3)$$

where Φ and θ are the parameters for the encoder and the decoder of the VAE model. The objective is equivalent to :

$$\mathcal{L}_{vae} = \mathbb{E}_{z\sim q_\theta(z|v)}[log\ p_\theta(z|v)] - \beta \mathbb{KL}(q_\Phi(z|v)||p(z)) \quad (4)$$

where β is the hyper-parameter which characterizes the pressure for the posterior $q_\Phi(z|v)$ to match Gaussian prior $p(z)$. The first term is the expectation of negative log-likelihood of instance v, referring to the reconstruction loss and leading to the preservation of the semantic information. The second term is a regularizer based on Kullback-Leibler divergence $\mathbb{KL}(\cdot)$ between the prior distribution $q_\theta(z|v)$ and the posterior distribution p_z. The prior is typically set to the isotropic unit Gaussian distribution $\mathcal{N}(0, 1)$.

We use multilayer perceptron (MLP) models (mainly containing two linear layers for dimension reduction and reconstruction) as the encoder and the decoder of the VAE model. Due to the reason that the sampled value z reduces the topology information of the training data and there exist latent variable collapse issues [27], here we adopt the feature extractor from the second last layer in the encoder for classification with parameters θ'. The binary cross-entropy loss is used as the training objective for classifying the relevance of the scientific document:

$$\mathcal{L}_{cls} = -\mathbb{E}[log\ p(y|v;\theta')]. \quad (5)$$

Overall, the training objective of the VAE-based document-level classification follows:

$$\mathcal{L}_{doc} = \mathcal{L}_{cls} + \mathcal{L}_{vae}. \tag{6}$$

Ranking with Trained Classifier. We use the trained classifier θ' for identifying the relevance of unlabeled data. For each document candidate d with the document representation v^{un}, we calculate the relevance score as follows:

$$s(d_i) = p(y = 1|v^{un}; \theta') \tag{7}$$

where $y = 1$ refers to the label *relevant*. The first ranking list $[r_d^1]$ is calculated based on the array of relevance score $[s(d)]$ in descending order.

4.3 Phrase-level Feature Classification

Our pilot user study suggests that a frequent clue for rejecting irrelevant documents is key phrase matching, which motivates our integration of phrase-level semantic features.

Phrase Selection We first want to select phrases that are more related to the relevant documents. Hence, we define the following two measures to select phrases:

Indicative: Ideally, a phrase that is indicative of relevant documents should be frequent in relevant documents. Therefore, we design our relevant-indicative measure as:

$$ID(p) = \frac{n_{p,1}}{|\{D_l \cap R\}|} \tag{8}$$

where $n_{p,x}$ is the number of labeled documents of relevance label x that p appears.

Unusual: Since the relevant documents only take up a small proportion of all document candidates, we want phrases that are unusual. To incorporate this, we design a measure of inverse document frequency:

$$UN(p) = \log \frac{|D_l|}{|n_{p,1} \cup n_{p,0}|}. \tag{9}$$

Inspiring by [36], we use geometric mean to combine these two measures, which provides a score for each phrase in the labeled documents. We rank all available phrases by this score and only select the top 30 % for the following steps.

Phrase Clustering & Feature Selection To capture the semantic relation between phrases, we construct a graph where each node on it represents a phrase. Edges are built according to the two nodes' semantic similarity w, which we calculate as the cosine similarity between their pre-trained MLM-based embeddings:

$$w_{i,j} = \sqrt{\max\left(\text{CosSim}(e_{p_i}, e_{p_j}), 0\right)}. \tag{10}$$

After constructing the phrase graph, we utilize an unsupervised community detection algorithm, Louvain clustering [7], to generate non-overlapping communities in this graph. The reason we choose Louvain over other clustering methods is that it does not require the number of clusters given ahead and can be used flexibly on the corpus from very different scientific domains.

After putting those phrases with similar semantics in a cluster, we continue to choose phrase-level features from these clusters based on the assumption that the phrase-level feature should have a stronger correlation with relevant documents. So for each cluster c_i, we count the number of positive documents it is related to (D_{c_i}), and a cluster is selected as a phrase-level feature if it is larger than α percent of positively labeled documents:

$$C_s = \{c_i | |D_{c_i}| > \alpha \cdot |D_l|\}, \tag{11}$$

$$D_{c_i} = \{d | d \in D_p, p \in c_i\}, \tag{12}$$

$$D_p = \{d | p \in d, d \in \{D_l \cap R\}\}. \tag{13}$$

Pseudo label generation. As the number of irrelevant documents is much larger than the relevant ones in the systematic review corpus, we can generate pseudo labels from unlabeled documents ($\mathcal{D}_u$) by using our trained VAE classifier. For each $d_i \in \mathcal{D}_u$, we calculate its probability to be relevant. Then we rank $\mathcal{D}_u$ based on this probability and select documents in the lowest 30% as pseudo-negative samples. This pseudo data is used together with the labeled documents $\mathcal{D}_l$ to train our phrase-level feature classifier.

Phrase-level Feature Classification To train this classifier, we need to calculate the corresponding value of the phrase-level feature for each training document. For a phrase mentioned in a training document, we first calculate the cosine similarity between the phrase and its feature cluster's centroid. Then the largest value from each cluster is set as the feature value:

$$f_{p,c_i} = \max\left(\text{CosSim}(e_p, e_{c_i})\right), \tag{14}$$

$$e_{c_i} = \sum w_p \cdot e_p\,, \quad w_p = \frac{|D_p|}{D_{c_i}}. \tag{15}$$

Then, for each phrase-level feature, we can calculate its corresponding value in the document as:

$$F_{d,j} = \max\left(\{f_{p,c_j}\} | p \in d, p \in c_j, c_j \in C_s\right). \tag{16}$$

With the phrase-level feature values F, the labeled documents D_l, and pseudo-labeled documents, we train a Random Forest model to learn which important phrase-level features matter for relevant documents. Finally, we use this trained model to predict and re-rank top k documents from the first ranking result $[r_d^1]$ to get the second-ranking list $[r_d^2]$ and perform the ranking ensemble process.

4.4 Ranking Ensemble

A relevant document should be ranked higher in both document-level ranking list $[r_d^1]$ and phrase-level ranking list $[r_d^2]$. Therefore, we use the ranking ensemble method on two ranking lists so that the highly-ranked irrelevant documents in the first ranking list can be rectified by our phrase-level feature classifier. For each unlabeled document candidate d, we calculate its final score by summing up its mean reciprocal rank scores in each ranking list:

$$mrr(d) = \sum_{t=1}^{2} \frac{1}{r_d^t} \tag{17}$$

where r_d^t is its ranking in the ranking list t.

SciMine ranks the final scores in descending order and return the first document for user screening according to the certainty-based query strategy. After the scholar reads and labels the document, this document is moved from $\mathcal{D}_u$ to $\mathcal{D}_l$. Then the scholar can decide whether he wants to stop SciMine or starts the next iteration.

Dataset	# Docs	# Pos	# Sents	# Words	# Phrases
Calcium	1069	246	11.6	288.9	38.5
Nudging	2019	101	11.2	281.1	31.4
Depression	1993	280	7.4	206.3	35.2
Virus	2481	120	8.8	219.5	36.9
AgriDiv	1505	129	10.1	275.4	43.4

Table 1: Datasets statistics. # Pos for the number of relevant documents; # Sents, # Words, # Phrases for the average number of sentences, words, and phrases in the documents.

5 EXPERIMENTS

5.1 Experimental Setup

Datasets. We conduct our experiments on four previously published datasets[2] and one newly created dataset. These datasets are from different research domains and the percentage of relevant documents ranges from 4.6% to 23.0%. Table 1 summarizes the statistics for them.

- **Calcium** [12]: This dataset is released in research on how to use citation classification to accelerate systematic review. The theme of this dataset is studying calcium channel blockers and it is in the medicine domain.
- **Nudging** [41]: This dataset is about a systematic review in the social science domain. The theme of this research is nudging healthcare professionals into evidence-based medicine.
- **Depression** [4]: This dataset is in the animal science domain and comprehensively includes published preclinical non-human animal literature on depression.
- **Virus** [23]: This dataset is from the medicine domain and is about performing viral Metagenomic Next-Generation Sequencing (mNGS) in common livestock.
- **AgriDiv**: In order to understand how scholars do a systematic review, we collaborate with experts in the ecological domain to create this dataset for their research. The theme of this research is to investigate the effect of agriculture diversification on rice production. We collect 1505 documents by searching the Web of Science and Scopus. Then two domain experts were invited to label the corpus and 129 studies are confirmed as relevant.

Compared Methods. We compare the following methods whose information includes lexical-level, sentence-level, and document-level features.

- TF-IDF+NB: We test this machine learning model with TF-IDF as the feature extraction and Naive Bayers as the classifier. According to [16, 54], this method is able to outperform models with more sophisticated feature information.
- D2V+SVM: We use doc2vec [24] as the feature extraction and the Support Vector Machine as the classifier.
- HierTrans: We use a pre-trained model SimCSE [17] to learn sentence embeddings in each document and utilize a hierarchical transformer as the classifier. By learning different weights of sentences in each training sample, this hierarchical model [36, 65] can achieve good performance in classification tasks with less training data.
- SPECTER-Once: Use the same document embedding that we gained during the preprocessing step. Train an SVM classifier with the initial seed set and predict the unlabeled studies to get a one-time retrieval result.
- SPECTER+SVM: This method utilizes the same document embedding that we gained during the prepossessing step and uses an SVM as a classifier.
- SPECTER+MLP: Use the same document embedding that we gained during the prepossessing step and use a 1-layer multilayer perceptron as the classifier.
- SciMine-NoPFC: An ablation of our framework that removes the phrase-level feature classification module.
- SciMine: Our proposed framework captures both document-level information and phrase-level information.

Implementation Details. For testing purposes, instead of screening the datasets by domain experts, we simulate the screening process by comparing the newly retrieved document to the gold label in each human-in-the-loop iteration. The simulation starts with a seed set of 5 relevant and 5 irrelevant studies and the classification model is retrained after the end of each iteration. The model is terminated once it has reached the target recall of relevant documents. We set this number to 0.95 in our case. The initial seed set is picked randomly from the corpus. For avoiding bias from the initial seed set, we create 5 seed sets for each dataset by randomly picking documents from the corpus. We test baseline methods and our proposed models based on these seed sets and every simulation is run 10 times for each seed set.

We utilize Adam with a weight decay rate of $1e-4$ to optimize our model. Except for TF-IDF+SVM and SPECTER, all baselines as well as SciMine are trained 200 rounds in each human-in-the-loop iteration. The learning rate is set to $1e-4$, the pressure β of VAE is 0.1, the batch size is 40, the a is 0.5 for selecting feature clusters, and the k is 50 for the ranking ensemble. We use the certainty-based query strategy, which retrieves the document with a high probability to be relevant from the prediction result.

Evaluation Metrics. We follow previous studies and evaluate our results using Work Saved over Sampling (WSS) and Relevant References Found (RRF). Given a level of recall, WSS calculates the reduction of documents needed to be screened. For instance, WSS@95 measures the percentage of records that can be saved when 95% of relevant documents have been identified by the user. Meanwhile, RRF@10 evaluates how many relevant documents can be identified when 10% of the unlabeled documents have been screened. It is used as a quick overview of the relevant documents.

5.2 Results

Table 2 shows the main results. In terms of WSS, the classic TF-IDF+NB model has very stable performance across datasets. As a method of capturing lexical features, it beats the D2V+SVM model in three datasets, which learns more sophisticated word embeddings. The Hierarchical Transformer model also does not perform very well on most of the datasets when compared to TF-IDF+NB and D2V+SVM. It may be because of the too-limited training data in our problem setting. By using the scientific literature-related document embedding, SPECTER+SVM and SPECTER+MLP outperform TF-IDF+NB, which demonstrates the advantage of richer semantic features over lexical features. It also proves that the advanced pre-trained language model can be applied to the task of

[2]https://github.com/asreview/systematic-review-datasets

Methods	Calcium			Nudging			Depression			Virus			AgriDiv		
	RRF@10	WSS@85	WSS@95	RRF@10	WSS@85	WSS@95	RRF@10	WSS@85	WSS@95	RRF@10	WSS@85	WSS@95	RRF@10	WSS@85	WSS@95
TF-IDF+NB	0.285	0.509	0.287	0.679	0.836	0.734	0.582	0.776	0.444	0.683	0.856	0.716	0.512	0.748	0.629
D2V+SVM	0.236	0.522	0.294	0.661	0.852	0.752	0.565	0.759	0.428	0.647	0.816	0.687	0.419	0.405	0.234
HierTrans	0.235	0.425	0.232	0.707	0.843	0.713	0.512	0.683	0.395	0.689	0.756	0.633	0.412	0.512	0.358
SPECTER+Once	0.220	0.301	0.185	0.622	0.782	0.694	0.486	0.654	0.406	0.494	0.693	0.509	0.308	0.598	0.442
SPECTER+SVM	0.259	0.560	0.291	0.724	0.827	0.743	0.553	0.806	0.646	0.745	0.861	0.710	0.545	0.736	0.653
SPECTER+MLP	0.268	0.612	0.316	0.723	0.872	**0.808**	0.596	0.836	0.678	0.766	0.870	0.742	0.522	0.782	0.686
SciMine-NoPFC	0.280	0.618	0.332	**0.739**	0.865	0.795	0.612	0.841	0.705	**0.782**	0.879	0.798	0.525	0.807	0.704
SciMine	**0.287**	**0.635**	**0.346**	0.735	**0.875**	0.802	**0.625**	**0.852**	**0.738**	0.780	**0.893**	**0.812**	**0.531**	**0.815**	**0.736**

Table 2: Evaluation results on five real-world datasets over the metric Work Saved over Sampling 85 percent and 95 percent of relevant documents (WSS@85 & WSS@95) and Relevant References found by the first 10% of iterations (RRF@10). For each dataset, models are tested on 5 randomly-sampled seed set to avoid bias.

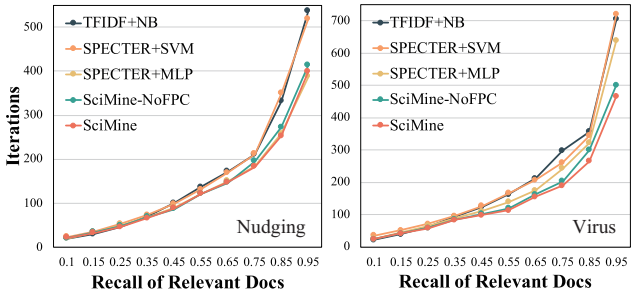

Figure 3: The visualization of the relevant document-finding process. The X-axis represents the number of retrieved relevant documents and the Y-axis is the number of iterations.

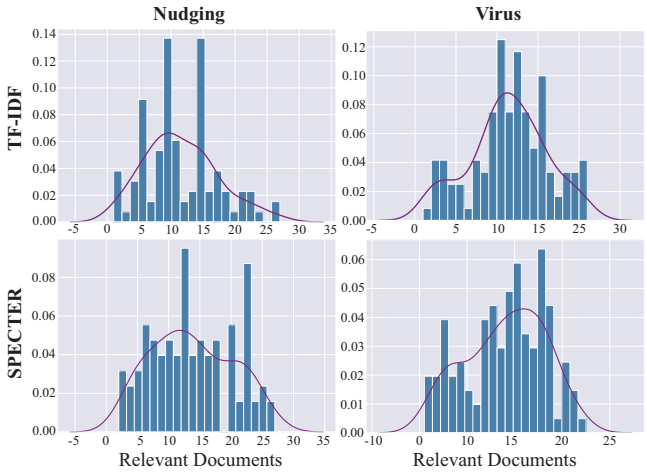

Figure 4: Density distribution of the relevant documents in Nudging and Virus through feature extraction methods such as TF-IDF, and SPECTER.

screening prioritization to boost the screening process. Not surprisingly, the SPECTER+Once works terribly, which indicates that even the SPECTER makes good document-level embedding on the scientific literature, only given the initial seed set is not sufficient and a human-in-the-loop session is a must. SciMine outperforms all baseline methods on most datasets by a large margin. It is consistently better than SciMine-NoPFC as well, which verifies that the document-level and phrase-level feature information are complementary. SciMine-NoPFC can outperform SPECTER+MLP on four datasets except **Nudging**, which demonstrates the necessity of an autoencoder to preserve the feature information. We further analyze the reason in Section 4.3.1.

For the measurement of RRF@10, we can see that the TF-IDF+NB can outperform several PLM-based models in **Calcium** and **Depression**. It indicates that the lexical feature is good at finding relevant documents during the early stage of the screening process.

We also visualize the process of finding the relevant documents on **Nudging** and **Virus** in Figure 3. We can see that SciMine can lead this competition in the whole process by finding more relevant documents in less human-in-the-loop iterations. And even though the lexical feature model performs well initially, the gap between the lexical feature model and other methods becomes obvious as the iteration goes on. In a real use case, when a user decides when to stop the model, this low efficiency by the lexical feature model may hinder the user from finding more relevant documents.

5.3 Further Analysis

5.3.1 Representation Analysis. We propose an intuitive method based on k nearest neighbors (kNN) to show the feature similarity of the documents. We search k nearest neighbors of each positive instance d_i through the Euclidean distance of the representations from encoding models such as TF-IDF, and SPECTER. Then we count the number of positive instances around d_i, formulated as:

$$Density(d_i) = \sum_{j}^{kNN(d_i)} \mathbf{1}_{y_j=1} \tag{18}$$

where j is the index of the k nearest neighbors of d_i. The density results are shown in Figure 4 with kernel distribution estimation (KDE) plot. First, as is observed in the density distributions of the relevant documents through TF-IDF representations, the peaks of the distribution curve are about 10 and 11 for **Nudging** and **Virus**, respectively. In comparison, in the density distribution through SPECTER representations, the density peaks further shift to larger ranges, 12 for **Nudging** and 17 for **Virus**, which implies that the features of relevant documents tend to become denser. The phenomenon demonstrates that the pre-trained language model SPECTER can obtain more informative features of the scientific documents than traditional methods such as TF-IDF, and as a result, the relevant documents can be more easily classified.

In SPECTER embeddings, the density distributions of relevant documents in **Virus** (the distribution of **AgriDiv**, and **Calcium** are

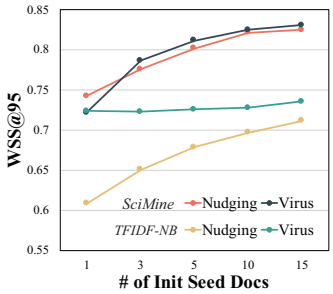

Figure 5: Parameter Study

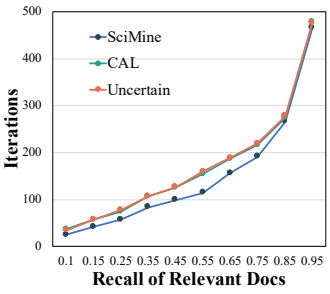

Figure 6: Query Strategy

shown in the Appendix) show that the representations of most relevant documents have significant feature similarity. The continuous distribution curves have peak shifts to large densities. It implies that most representations of the relevant documents have typical features for the CE-trained classifier (with only an MLP feature extractor) to distinguish, but there exist some documents having few similar features, difficult for such a classifier due to feature bias. The phenomenon also aligns with proposition 1 in Section 4.3. But for **Nudging**, the density distribution is relatively uniform compared with other ones, which implies the feature bias is less significant and a classifier with the MLP feature extractor can also achieve great performance. The distribution explains the reason that SPECTER+MLP achieves stronger WSS@95 compared with SciMine-NoPFC on **Nudging**, but fails on others.

5.3.2 Parameter Studies. We experiment to understand how varying the number of documents in the initial seed set influences the performance of our model. For each dataset, we create a seed set by randomly sampling n relevant and n irrelevant documents. We vary n from 1 to 15 and plot the results in Figure 4. We can see that on the **Nudging** dataset, the performance of our model improves significantly when $n < 10$ and gradually saturates when $n \geq 10$. A similar trend can be observed on the **Virus** dataset, as shown in Figure 5. This verifies that our model only needs around 10 documents in total to achieve reasonable performance, which is affordable for most scholars. It is also interesting to notice that TF-IDF+NB achieves comparable results with SciMine on **Virus** dataset when n is 1, but the gap between the two methods becomes obvious as n increases.

5.3.3 The Influence of Query Strategy. Query strategy decides how the active learner retrieves the document from the predictions for human annotation. The most widely used query strategy applied to our task is certainty-based, which selects the document with the highest probability of being relevant. The other common strategy used is uncertainty-based, which selects the "hard relevant" document. Recently, one kind of uncertainty-based query strategy called Contrastive Active Learning (CAL) becomes popular in the Active Learning area[35, 66]. This strategy tries to pick the most contrastive example, for instance, the probability of it and its neighbors' having the largest Kullback-Leibler divergence. We test how the query strategy influences the performance of our model.

As shown in Figure 6, SciMine with the certainty-based strategy still performs best. We can also observe that even if the two uncertainty-based models have some difficulties finding relevant documents in the early stage, it becomes more efficient during the second half. Regarding the final WSS@95, the margin between the certainty-based and two uncertainty-based models is not that large, which shows that our ranking model is robust to different query strategies.

Phrase-level Features	TF-IDF Features
Cluster1: {pig, sheep, cattle, goat, piglet…} Cluster2: {metagenomic analysis, pcr analysis, sequencing, deep sequencing … Cluster 3: {virus, posavirus, pestivirus …}	{'strain', 'genotype', 'piglet', 'human', 'identified', 'fecal sample', 'metagenomic', 'bovine', 'virus'…}

Document Candidate

"…next generation sequencing research tool hand helping explore unknown field human veterinary virology . Metagenomic analysis enabled discovery putative novel pathogen identification etiologic agent disease, solving long standing mystery caused divergent virus. Approach study investigating fecal sample..."

Figure 7: One example of an irrelevant document with SciMine discovered Document-level features (rectangled with red), Phrase-level features (colored with orange), and TF-IDF's lexical features (underlined green).

5.3.4 Case Study. Figure 7 shows one irrelevant document from the **Virus** dataset whose research theme is "performing viral Metagenomic Next-Generation Sequencing (mNGS) in common livestock". We also list the phrase-level features that SciMine discovered and the lexical features the TF-IDF model relies on. It can be seen that TF-IDF can recognize some important phrases like "piglet", "virus" and "metagenomic". However, it also weighs on some spurious phrases like "human" and "identified". These two phrases may appear more in the labeled relevant documents but do not indeed imply relevance. In contrast, SciMine detects three clusters of phrase-level features, which are related to livestock, sequencing approach, and virus, separately. These clusters are also in accord with the research theme. Therefore, when the candidate document does not mention anything related to common livestock, SciMine ranks this document lower. We also apply the SCD method [22] to highlight features that the VAE model in SciMine discovers. VAE weighs on important features like "next generation sequencing" and "virus" while skipping spurious patterns like "fecal sample".

5.3.5 User Study. To understand how real end users experience SciMine, we perform a user study. We design a UI interface for screening prioritization models. Six Ph.D. students in the ecological domain were invited to join the study. They were divided into two groups: three students used SciMine while the other three used the TF-IDF+SVM model. Before the test began, they were asked to fully understand the research theme of **AgriDiv** and were instructed to know how to label the relevant/irrelevant documents. Furthermore, they were informed that they could stop the model whenever they felt there were no remaining relevant documents left. As a result, the average recall for SciMine and TFIDF+SVM is 91.3% and 83.7 %, which demonstrates the effectiveness of SciMine in the real screening scenario. However, both scores are lower than

0.95, which suggests that the traditional WSS score may not truly reflect the performance of the model. Users tend to stop earlier when the model constantly recommends irrelevant documents to them.

6 CONCLUSION

We proposed SciMine, a novel human-in-the-loop framework for efficient screening prioritization. Different from previous methods that solely rely on lexical information, we study how to apply the contextual information from pre-trained language models for this task. SciMine captures two types of information: document-level and phrase-level from the corpus and uses rank ensemble to finalize the prediction. To understand how scholars work in a systematic review, we contribute a dataset **AgriDiv** in the ecological domain. Experiments on five real-world datasets show that the richer semantic features are useful for the screening prioritization task since SciMine framework allows rich pre-trained knowledge to outperform discrete token features, achieving state-of-the-art results across 5 benchmarks, and providing analysis using different feature extractions. We conclude that: (1) The classic lexical-based methods may result in feature bias; (2) Feature bias can easily occur in the task of iterative screening prioritization; (3) Contextualized document-level and phrase-level information are complementary in solving feature bias for this task. In the future, we plan to extend our framework by allowing models to incorporate more user-provided information. For example, the human rationale in text patterns can be used to teach the model in each iteration or provides a sentence describing his research theme as another seed of information.

ACKNOWLEDGEMENT

Yue Zhang is the corresponding author. We would like to thank the anonymous reviewers for the detailed and thoughtful reviews. The work is funded by the Pioneer and "Leading Goose" R&D Program of Zhejiang under Grant Number 2022SDXHDX0003.

A APPENDIX

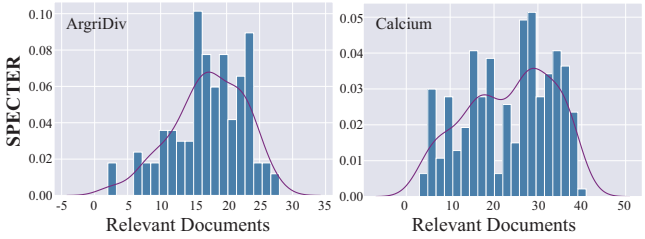

Figure 8: Density distribution of the relevant documents in AgriDiv and Calcium through the feature extraction method SPECTER.

A.1 Empirical Study for Feature Bias

To demonstrate the feature bias of active learning, we illustrate the representations of dataset **Virus** from the trained feature extractor by using t-SNE. We show the representations in Figure 9 when

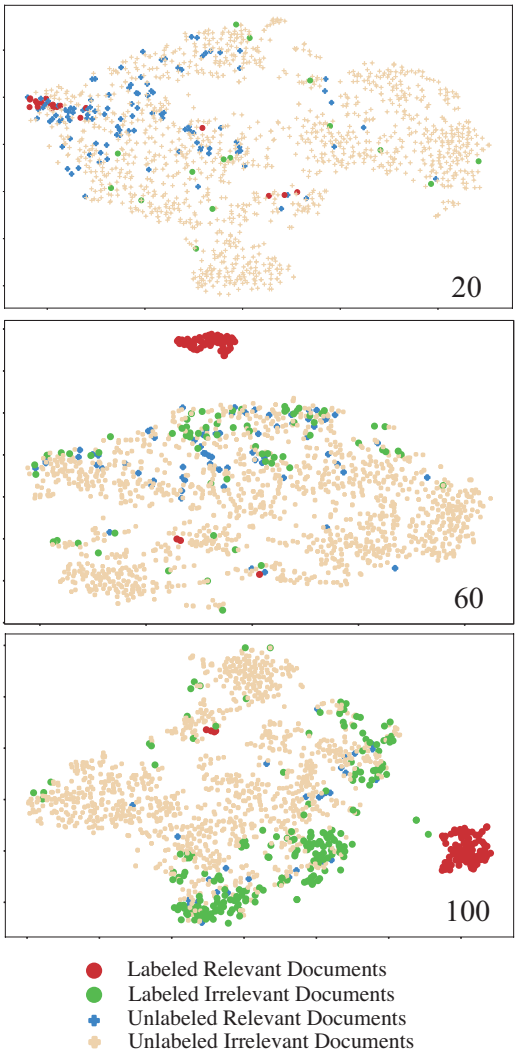

Figure 9: Visualization of document representations obtained from the MLP feature extractor. We use t-SNE to transfer the feature space into two-dimensional space.

the labeled relevant documents are 20, 60, and 100, respectively. Obviously, in the sub-figure 60 and 100, the labeled documents cluster together for feature bias, but other semantic information is overlooked and these unlabeled irrelevant data locate far away from labeled relevant data.

A.2 Density Distribution of Relevant Documents

We also show the density distributions of relevant documents through SPECTER embeddings using Eq(1) in Figure 8. The density distributions are similar to that of **Virus**, in that the peak shifts to a large density. It indicates that there exist a small proportion of relevant documents having few similar features to the majority, which are difficult for the minimal CE loss trained classifier with an MLP feature extractor.

REFERENCES

[1] Amal Alharbi and Mark Stevenson. 2019. Ranking studies for systematic reviews using query adaptation: University of Sheffield's approach to CLEF eHealth 2019 task 2 working notes for CLEF 2019. In *Working Notes of CLEF 2019-Conference and Labs of the Evaluation Forum*, Vol. 2380. CEUR Workshop Proceedings.

[2] Zafar Ali, Guilin Qi, Pavlos Kefalas, Shah Khusro, Inayat Khan, and Khan Muhammad. 2022. SPR-SMN: scientific paper recommendation employing SPECTER with memory network. *Scientometrics* (2022), 1–23.

[3] Jordan T Ash, Chicheng Zhang, Akshay Krishnamurthy, John Langford, and Alekh Agarwal. 2019. Deep batch active learning by diverse, uncertain gradient lower bounds. *arXiv preprint arXiv:1906.03671* (2019).

[4] Alexandra Bannach-Brown, Jing Liao, Gregers Wegener, and Malcolm Macleod. 2016. Understanding in vivo modelling of depression in non-human animals: a systematic review protocol. *Evidence-based Preclinical Medicine* 3, 2 (2016), 20–27.

[5] Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. SciBERT: A pretrained language model for scientific text. *arXiv preprint arXiv:1903.10676* (2019).

[6] Chandra Bhagavatula, Sergey Feldman, Russell Power, and Waleed Ammar. 2018. Content-based citation recommendation. *arXiv preprint arXiv:1802.08301* (2018).

[7] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. 2008. Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment* 2008, 10 (2008), P10008.

[8] Hong-You Chen, Cheng-Syuan Lee, Keng-Te Liao, and Shou-De Lin. 2018. Word Relation Autoencoder for Unseen Hypernym Extraction Using Word Embeddings. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Brussels, Belgium, 4834–4839. https://doi.org/10.18653/v1/D18-1519

[9] Jiaao Chen, Zichao Yang, and Diyi Yang. 2020. Mixtext: Linguistically-informed interpolation of hidden space for semi-supervised text classification. *arXiv preprint arXiv:2004.12239* (2020).

[10] SH Cheng, C Augustin, A Bethel, D Gill, S Anzaroot, J Brun, B DeWilde, RC Minnich, R Garside, YJ Masuda, et al. 2018. Using machine learning to advance synthesis and use of conservation and environmental evidence. (2018).

[11] Arman Cohan, Sergey Feldman, Iz Beltagy, Doug Downey, and Daniel S Weld. 2020. Specter: Document-level representation learning using citation-informed transformers. *arXiv preprint arXiv:2004.07180* (2020).

[12] Aaron M Cohen, William R Hersh, Kim Peterson, and Po-Yin Yen. 2006. Reducing workload in systematic review preparation using automated citation classification. *Journal of the American Medical Informatics Association* 13, 2 (2006), 206–219.

[13] Gordon V Cormack and Maura R Grossman. 2015. Autonomy and reliability of continuous active learning for technology-assisted review. *arXiv preprint arXiv:1504.06868* (2015).

[14] Gordon V Cormack and Maura R Grossman. 2016. Scalability of continuous active learning for reliable high-recall text classification. In *Proceedings of the 25th ACM international on conference on information and knowledge management*. 1039–1048.

[15] Liat Ein Dor, Alon Halfon, Ariel Gera, Eyal Shnarch, Lena Dankin, Leshem Choshen, Marina Danilevsky, Ranit Aharonov, Yoav Katz, and Noam Slonim. 2020. Active learning for BERT: an empirical study. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 7949–7962.

[16] Gerbrich Ferdinands, Raoul Schram, Jonathan de Bruin, Ayoub Bagheri, Daniel Leonard Oberski, Lars Tummers, Rens van de Schoot, et al. 2020. Active learning for screening prioritization in systematic reviews-A simulation study. (2020).

[17] Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple Contrastive Learning of Sentence Embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Online and Punta Cana, Dominican Republic, 6894–6910. https://doi.org/10.18653/v1/2021.emnlp-main.552

[18] Maura R Grossman, Gordon V Cormack, and Adam Roegiest. 2017. Automatic and semi-automatic document selection for technology-assisted review. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 905–908.

[19] Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel R Bowman, and Noah A Smith. 2018. Annotation artifacts in natural language inference data. *arXiv preprint arXiv:1803.02324* (2018).

[20] Md Reshad Ul Hoque, Jiang Li, and Jian Wu. 2022. SciEv: Finding Scientific Evidence Papers for Scientific News. *arXiv preprint arXiv:2205.00126* (2022).

[21] Kexin Huang, Jaan Altosaar, and Rajesh Ranganath. 2019. Clinicalbert: Modeling clinical notes and predicting hospital readmission. *arXiv preprint arXiv:1904.05342* (2019).

[22] Xisen Jin, Zhongyu Wei, Junyi Du, Xiangyang Xue, and Xiang Ren. 2019. Towards hierarchical importance attribution: Explaining compositional semantics for neural sequence models. *arXiv preprint arXiv:1911.06194* (2019).

[23] Kirsty TT Kwok, David F Nieuwenhuijse, My VT Phan, and Marion PG Koopmans. 2020. Virus metagenomics in farm animals: a systematic review. *Viruses* 12, 1 (2020), 107.

[24] Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *International conference on machine learning*. PMLR, 1188–1196.

[25] Grace E Lee and Aixin Sun. 2018. Seed-driven document ranking for systematic reviews in evidence-based medicine. In *The 41st international ACM SIGIR conference on research & development in information retrieval*. 455–464.

[26] Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2020. BioBERT: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics* 36, 4 (2020), 1234–1240.

[27] Ruizhe Li, Xiao Li, Chenghua Lin, Matthew Collinson, and Rui Mao. 2019. A Stable Variational Autoencoder for Text Modelling. In *Proceedings of the 12th International Conference on Natural Language Generation*. Association for Computational Linguistics, Tokyo, Japan, 594–599. https://doi.org/10.18653/v1/W19-8673

[28] Han Liu, Xiaotong Zhang, Lu Fan, Xuandi Fu, Qimai Li, Xiao-Ming Wu, and Albert YS Lam. 2019. Reconstructing capsule networks for zero-shot intent classification. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. 4799–4809.

[29] Jialu Liu, Jingbo Shang, Chi Wang, Xiang Ren, and Jiawei Han. 2015. Mining quality phrases from massive text corpora. In *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*. 1729–1744.

[30] Jinghui Lu, Linyi Yang, Brian Namee, and Yue Zhang. 2022. A Rationale-Centric Framework for Human-in-the-loop Machine Learning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 6986–6996.

[31] Yun Luo, Zihan Liu, Yuefeng Shi, Stan Z Li, and Yue Zhang. 2022. Exploiting Sentiment and Common Sense for Zero-shot Stance Detection. *arXiv preprint arXiv:2208.08797* (2022).

[32] Kelvin Luu, Xinyi Wu, Rik Koncel-Kedziorski, Kyle Lo, Isabel Cachola, and Noah A Smith. 2020. Explaining relationships between scientific documents. *arXiv preprint arXiv:2002.00317* (2020).

[33] Shuming Ma, Xu Sun, Junyang Lin, and Houfeng Wang. 2018. Autoencoder as Assistant Supervisor: Improving Text Representation for Chinese Social Media Text Summarization. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Association for Computational Linguistics, Melbourne, Australia, 725–731. https://doi.org/10.18653/v1/P18-2115

[34] Sean MacAvaney, Arman Cohan, and Nazli Goharian. 2020. SLEDGE-Z: A zero-shot baseline for COVID-19 literature search. *arXiv preprint arXiv:2010.05987* (2020).

[35] Katerina Margatina, Giorgos Vernikos, Loïc Barrault, and Nikolaos Aletras. 2021. Active learning by acquiring contrastive examples. *arXiv preprint arXiv:2109.03764* (2021).

[36] Dheeraj Mekala and Jingbo Shang. 2020. Contextualized weak supervision for text classification. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 323–333.

[37] Yu Meng, Jiaming Shen, Chao Zhang, and Jiawei Han. 2018. Weakly-supervised neural text classification. In *proceedings of the 27th ACM International Conference on information and knowledge management*. 983–992.

[38] Yu Meng, Jiaming Shen, Chao Zhang, and Jiawei Han. 2019. Weakly-supervised hierarchical text classification. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 33. 6826–6833.

[39] Yu Meng, Yunyi Zhang, Jiaxin Huang, Chenyan Xiong, Heng Ji, Chao Zhang, and Jiawei Han. 2020. Text classification using label names only: A language model self-training approach. *arXiv preprint arXiv:2010.07245* (2020).

[40] Takeru Miyato, Andrew M Dai, and Ian Goodfellow. 2016. Adversarial training methods for semi-supervised text classification. *arXiv preprint arXiv:1605.07725* (2016).

[41] Rosanna Nagtegaal, Lars Tummers, Mirko Noordegraaf, Victor Bekkers, et al. 2019. Nudging healthcare professionals towards evidence-based medicine: a systematic scoping review. *Journal of Behavioral Public Administration* 2, 2 (2019).

[42] Malte Ostendorff, Nils Rethmeier, Isabelle Augenstein, Bela Gipp, and Georg Rehm. 2022. Neighborhood contrastive learning for scientific document representations with citation embeddings. *arXiv preprint arXiv:2202.06671* (2022).

[43] Malte Ostendorff, Terry Ruas, Till Blume, Bela Gipp, and Georg Rehm. 2020. Aspect-based document similarity for research papers. *arXiv preprint arXiv:2010.06395* (2020).

[44] Mourad Ouzzani, Hossam Hammady, Zbys Fedorowicz, and Ahmed Elmagarmid. 2016. Rayyan—a web and mobile app for systematic reviews. *Systematic reviews* 5, 1 (2016), 1–10.

[45] Ameya Prabhu, Charles Dognin, and Maneesh Singh. 2019. Sampling bias in deep active classification: An empirical study. *arXiv preprint arXiv:1909.09389* (2019).

[46] Piotr Przybyla, Austin J Brockmeier, Georgios Kontonatsios, Marie-Annick Le Pogam, John McNaught, Erik von Elm, Kay Nolan, and Sophia Ananiadou. 2018. Prioritising references for systematic reviews with RobotAnalyst: a user

study. *Research synthesis methods* 9, 3 (2018), 470–488.

[47] Laila Rasmy, Yang Xiang, Ziqian Xie, Cui Tao, and Degui Zhi. 2021. Med-BERT: pretrained contextualized embeddings on large-scale structured electronic health records for disease prediction. *NPJ digital medicine* 4, 1 (2021), 1–13.

[48] Xiang Ren, Jialu Liu, Xiao Yu, Urvashi Khandelwal, Quanquan Gu, Lidan Wang, and Jiawei Han. 2014. Cluscite: Effective citation recommendation by information network-based clustering. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. 821–830.

[49] Harrisen Scells, Guido Zuccon, and Bevan Koopman. 2020. You can teach an old dog new tricks: Rank fusion applied to coordination level matching for ranking in systematic reviews. In *Advances in Information Retrieval: 42nd European Conference on IR Research, ECIR 2020, Lisbon, Portugal, April 14–17, 2020, Proceedings, Part I 42*. Springer, 399–414.

[50] Harrisen Scells, Guido Zuccon, Bevan Koopman, Anthony Deacon, Leif Azzopardi, and Shlomo Geva. 2017. Integrating the framing of clinical questions via PICO into the retrieval of medical literature for systematic reviews. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. 2291–2294.

[51] Christopher Schröder and Andreas Niekler. 2020. A survey of active learning for text classification using deep neural networks. *arXiv preprint arXiv:2008.07267* (2020).

[52] Jingbo Shang, Jialu Liu, Meng Jiang, Xiang Ren, Clare R Voss, and Jiawei Han. 2018. Automated phrase mining from massive text corpora. *IEEE Transactions on Knowledge and Data Engineering* 30, 10 (2018), 1825–1837.

[53] Megha Srivastava, Tatsunori Hashimoto, and Percy Liang. 2020. Robustness to spurious correlations via human annotations. In *International Conference on Machine Learning*. PMLR, 9109–9119.

[54] Rens van de Schoot, Jonathan de Bruin, Raoul Schram, Parisa Zahedi, Jan de Boer, Felix Weijdema, Bianca Kramer, Martijn Huijts, Maarten Hoogerwerf, Gerbrich Ferdinands, et al. 2021. An open source machine learning framework for efficient and transparent systematic reviews. *Nature Machine Intelligence* 3, 2 (2021), 125–133.

[55] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30 (2017).

[56] Byron C Wallace, Kevin Small, Carla E Brodley, Joseph Lau, and Thomas A Trikalinos. 2012. Deploying an interactive machine learning system in an evidence-based practice center: abstrackr. In *Proceedings of the 2nd ACM SIGHIT international health informatics symposium*. 819–824.

[57] Shuai Wang, Harrisen Scells, Justin Clark, Bevan Koopman, and Guido Zuccon. 2022. From little things big things grow: A collection with seed studies for medical systematic review literature search. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 3176–3186.

[58] Shuai Wang, Harrisen Scells, Bevan Koopman, and Guido Zuccon. 2022. Neural Rankers for Effective Screening Prioritisation in Medical Systematic Review Literature Search. *arXiv preprint arXiv:2212.09017* (2022).

[59] Shuai Wang, Harrisen Scells, Ahmed Mourad, and Guido Zuccon. 2022. Seed-driven document ranking for systematic reviews: A reproducibility study. In *European Conference on Information Retrieval*. Springer, 686–700.

[60] Yidong Wang, Hao Chen, Yue Fan, Wang Sun, Ran Tao, Wenxin Hou, Renjie Wang, Linyi Yang, Zhi Zhou, Lan-Zhe Guo, et al. 2022. USB: A Unified Semi-supervised Learning Benchmark. *arXiv preprint arXiv:2208.07204* (2022).

[61] Zihan Wang, Dheeraj Mekala, and Jingbo Shang. 2020. X-class: Text classification with extremely weak supervision. *arXiv preprint arXiv:2010.12794* (2020).

[62] Huaying Wu, Tingting Wang, Jiayi Chen, Su Chen, Qinmin Hu, and Liang He. 2018. Ecnu at 2018 ehealth task 2: Technologically assisted reviews in empirical medicine. *Methods* 4, 5 (2018), 7.

[63] Qizhe Xie, Zihang Dai, Eduard Hovy, Thang Luong, and Quoc Le. 2020. Unsupervised data augmentation for consistency training. *Advances in Neural Information Processing Systems* 33 (2020), 6256–6268.

[64] Eugene Yang, Sean MacAvaney, David D Lewis, and Ophir Frieder. 2022. Goldilocks: Just-right tuning of bert for technology-assisted review. In *Advances in Information Retrieval: 44th European Conference on IR Research, ECIR 2022, Stavanger, Norway, April 10–14, 2022, Proceedings, Part I*. Springer, 502–517.

[65] Linyi Yang, Tin Lok James Ng, Barry Smyth, and Riuhai Dong. 2020. Html: Hierarchical transformer-based multi-task learning for volatility prediction. In *Proceedings of The Web Conference 2020*. 441–451.

[66] Yue Yu, Lingkai Kong, Jieyu Zhang, Rongzhi Zhang, and Chao Zhang. 2022. AcTune: Uncertainty-Based Active Self-Training for Active Fine-Tuning of Pretrained Language Models. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 1422–1436.

[67] Zhe Yu, Nicholas A Kraft, and Tim Menzies. 2018. Finding better active learners for faster literature reviews. *Empirical Software Engineering* 23, 6 (2018), 3161–3186.

[68] Chenhan Yuan and Hoda Eldardiry. 2021. Unsupervised Relation Extraction: A Variational Autoencoder Approach. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Online and Punta Cana, Dominican Republic, 1929–1938. https://doi.org/10.18653/v1/2021.emnlp-main.147

[69] Michelle Yuan, Hsuan-Tien Lin, and Jordan Boyd-Graber. 2020. Cold-start active learning through self-supervised language modeling. *arXiv preprint arXiv:2010.09535* (2020).

[70] Jingqing Zhang, Piyawat Lertvittayakumjorn, and Yike Guo. 2019. Integrating semantic knowledge to tackle zero-shot text classification. *arXiv preprint arXiv:1903.12626* (2019).

[71] Lan Zhang, Wray Buntine, and Ehsan Shareghi. 2022. On the Effect of Isotropy on VAE Representations of Text. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Association for Computational Linguistics, Dublin, Ireland, 694–701. https://doi.org/10.18653/v1/2022.acl-short.78

[72] Jie Zou, Dan Li, and Evangelos Kanoulas. 2018. Technology assisted reviews: Finding the last few relevant documents by asking yes/no questions to reviewers. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. 949–952.