AI

MDPI

Article

Rule-Enhanced Active Learning for Semi-Automated Weak Supervision

David Kartchner [1,2,3], Davi Nakajima An [1,2], Wendi Ren [2], Chao Zhang [2,4] and Cassie S. Mitchell [1,3,4,*]

[1] Laboratory for Pathology Dynamics, Georgia Institute of Technology and Emory University, Atlanta, GA 30332, USA; david.kartchner@gatech.edu (D.K.); dna@gatech.edu (D.N.A.)
[2] School of Computer Science, Georgia Institute of Technology, Atlanta, GA 30332, USA; wren44@gatech.edu (W.R.); chaozhang@gatech.edu (C.Z.)
[3] Department of Biomedical Engineering, Georgia Institute of Technology and Emory University, Atlanta, GA 30332, USA
[4] Machine Learning Center at Georgia Tech, Georgia Institute of Technology, Atlanta, GA 30332, USA
[*] Correspondence: cassie.mitchell@bme.gatech.edu

Abstract: A major bottleneck preventing the extension of deep learning systems to new domains is the prohibitive cost of acquiring sufficient training labels. Alternatives such as weak supervision, active learning, and fine-tuning of pretrained models reduce this burden but require substantial human input to select a highly informative subset of instances or to curate labeling functions. REGAL (Rule-Enhanced Generative Active Learning) is an improved framework for weakly supervised text classification that performs active learning over labeling functions rather than individual instances. REGAL interactively creates high-quality labeling patterns from raw text, enabling a single annotator to accurately label an entire dataset after initialization with three keywords for each class. Experiments demonstrate that REGAL extracts up to 3 times as many high-accuracy labeling functions from text as current state-of-the-art methods for interactive weak supervision, enabling REGAL to dramatically reduce the annotation burden of writing labeling functions for weak supervision. Statistical analysis reveals REGAL performs equal or significantly better than interactive weak supervision for five of six commonly used natural language processing (NLP) baseline datasets.

Keywords: weak supervision; active learning; natural language processing; text classification; text mining; data labeling

check for updates

Citation: Kartchner, D.; Nakajima An, D.; Ren, W.; Zhang, C.; Mitchell, C.S. Rule-Enhanced Active Learning for Semi-Automated Weak Supervision. *AI* **2022**, *3*, 211–228. https://doi.org/10.3390/ai3010013

Academic Editors: Emma Tonkin, Kristina Yordanova and Rüdiger Buchkremer

Received: 17 December 2021
Accepted: 11 March 2022
Published: 16 March 2022

1. Introduction

Collecting training labels is a necessary, fundamental hurdle in creating any supervised machine learning system. The cost of curating labels, however, can be very costly. Training a robust deep learning model generally requires on the order of 10,000+ training examples [1,2]. Recently, advances in unsupervised pretraining [3–5] have created expressive, publicly available models with smaller requirements for task adaptation via fine tuning. Pretrained models in specific domains (e.g., clinical, biomedical, legal), refs. [6,7] have extended benefits to new domains.

Researchers have improved automated text class labeling using solutions that include active learning [8], domain adaptation of pretrained models, self-training on confident model predictions [9–11], noisy supervision using labeling heuristics [12], and crowd-sourced labeling [13].

Active learning [14] seeks to reduce the labeling burden by initializing a model with a very small set of seed labels, then iteratively solicits batches of labels on "highly informative" unlabeled instances. Active learning allows a model to be robustly trained on a small subset of data while attaining performance similar to a model trained on a much larger dataset. While active learning provides large gains compared to random instance labeling, significant work is still required to label individual data instances.

Weak supervision provides multiple overlapping supervision sources in the form of independent labeling rules, then probabilistically disambiguates sources to obtain predictions. Since a single labeling rule (also called a labeling function) can label a large proportion of a dataset, a small number of labeling rules can lead to significant gains in efficiency while minimizing annotation efforts. The main difficulty in weak supervision is the need to curate these labeling functions, which can be deceptively complex and nuanced.

To address limitations of prior labeling methods, we synthesize ideas from active learning, pretraining, and weak supervision to create REGAL, which performs active learning over model-generated labeling functions. REGAL accelerates data labeling by interactively soliciting human feedback on labeling functions instead of individual data points. It accomplishes this by (1) extracting high-confidence labeling rules from input documents, (2) soliciting labels on these proposed rules from a human user, and (3) denoising overlap between chosen labeling functions to create high-confidence labels. This framework, depicted in Figure 1 enables REGAL to seek feedback on areas of model weakness while simultaneously labeling large swaths of examples.

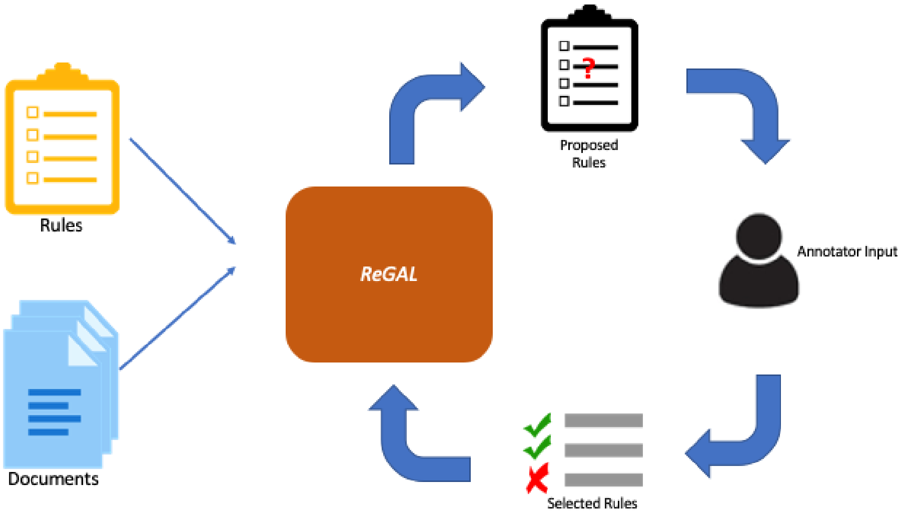


Figure 1. REGAL model setup. REGAL takes unlabeled documents and seed rules as input. It then iteratively proposes new labeling functions by extracting high-quality patterns from the training data and soliciting user feedback about which to keep.

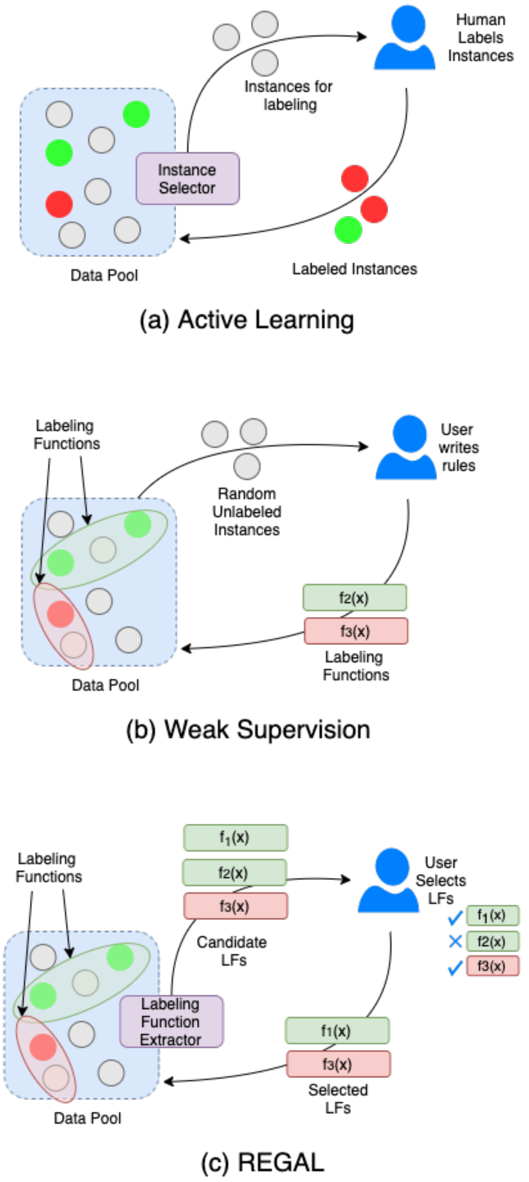
2. Preliminaries

REGAL proposes multiple, high-quality sources of weak supervision to improve labeling on a source dataset as formally defined below. Figure 2 illustrates the differences between active learning, weak supervision, and REGAL.

2.1. Problem Formulation

It is assumed that for a given a set of documents $\mathcal{D} = \{d_1, d_2, \ldots, d_{|D|}\}$, each of which has a (possibly unknown) classification label $c_i \in \mathcal{C}$. Each document $d_i = [v_{i,1}, v_{i,2}, \ldots, v_{i,T}]$ represents a sequence of tokens from the vocabulary $\mathcal{V}$, where tokens drawn from $\mathcal{V}$ could be words, subwords, characters, etc.

It is assumed there is no access to ground-truth labels for the documents in the training set. However, there are a small number of heuristic labeling functions (LFs) given which provide limited initial supervision for each class. $\mathcal{R} = \{r_1, r_2, \ldots, r_l\}$, where each $r_j : \mathcal{D} \to \mathcal{C} \cup \{c_{abstain}\}$ is a function that maps documents to a class label in $\mathcal{C}$ or abstains from labeling. This set of LFs induces a vector of noisy labels for each document, denoted $\pounds_i = [r_1(d_i), r_2(d_i), \ldots, r_l(d_i)]^T$. Because LFs act as rule-based labelers, we freely interchange the terms "labeling function" and "rule" throughout the paper.



(a) Active Learning

(b) Weak Supervision

(c) REGAL

Figure 2. Labeling structure for traditional active learning, weak supervision, and REGAL. In traditional active learning, high-value instances are selected and sent to a human annotators for labeling. In traditional weak supervision, annotators write rules based on patterns they observe in data. REGAL synthesizes these two approaches by extracting high-value candidate LFs which are then filtered by human annotators.

2.2. Challenges

Weakly supervised text classification presents three main challenges: label noise, label incompleteness, and annotator effort. For a lengthier discussion of different sources of label noise and the different types of algorithms used to address label incompleteness, see [15].

2.2.1. Label Noise

Label noise is the problem of labeling functions generating incorrect labels for particular data instances. This problem generally occurs when a specified labeling function is too general and, thus, mislabels instances into the wrong class. The diversity of language presents an extremely large space of possible misapplications for a single labeling function and enumerating these misapplications can be prohibitively expensive.

Recent works to address label noise include: generative label denoising [12,16], self-training on synthetic examples generated with latent variable models [17], using a neural network to identify improper applications of labeling functions using labeled rule exemplars [18], and active learning on instances with conflicting labels [19]. REGAL seeks to reduce label noise by automatically learning rules designed to differentiate between separate classes.

2.2.2. Label Incompleteness

Label incompleteness is the insufficiency of labeling functions to assign labels to particular slices of a dataset. It occurs when the syntactic and semantic patterns in a subset of examples do not lie within the scope of the given labeling functions. Label incompleteness is particularly pervasive in the long tails of a dataset, which often contain more diverse, difficult instances. For this reason, label incompleteness commonly manifests in low-resource or highly technical domains where differences in nomenclature lead to large labeling gaps.

Approaches to tackle label incompletness include differentiable soft-matching of labeling rules to unlabeled instances [20], automatic rule generation using pre-specified rule patterns [21,22], co-training a rule-based labeling module with a deep learning module capable of matching unlabeled instances [11,17], and encouraging LF diversity by interactively soliciting LFs for unlabeled instances [19].

2.2.3. Annotator Effort

Many domains require subject matter experts (SMEs) to annotate correctly. However, SMEs have cost and time constraints. These constraints are often most pressing in domains requiring the most expertise (e.g., biomedical), which is precisely where expert input is most valuable. By presenting annotators with candidate labeling rules, REGAL reduces the time necessary to specify rules by hand, thereby increasing annotator efficiency.

2.3. Objectives

REGAL is a model that interactively generates labeling functions from a text corpus with a small set of sparse, noisy labels. REGAL addresses text labeling challenges by automatically proposing labeling rules designed to (1) disambiguate instances with conflicting LF-induced labels and (2) extend coverage to unlabeled portions of the dataset. As annotators generate labels, REGAL can adapt to new labeling needs as the set of labels expands.

3. Methods

REGAL's architecture is shown in Figure 3. REGAL is composed of four components:

1. `TextEncoder`: Encodes semantically meaningful information about each token and document into contextualized token embeddings.
2. `SnippetSelector`: Extracts relevant phrases for document classification.
3. `RuleProposer`: Generates candidate labeling functions from extracted snippets.
4. `RuleDenoiser`: Produces probabilistic labels for all documents using labeling functions and document embeddings.

3.1. Text Encoder

REGAL begins with a `TextEncoder` module whose purpose is to create semantically meaningful embeddings for each document and its individual tokens. Token embeddings are used by the snippet selector and rule proposer to construct meaningful rules. The document embeddings are used by the rule denoiser to weight LF relevance for individual instances.

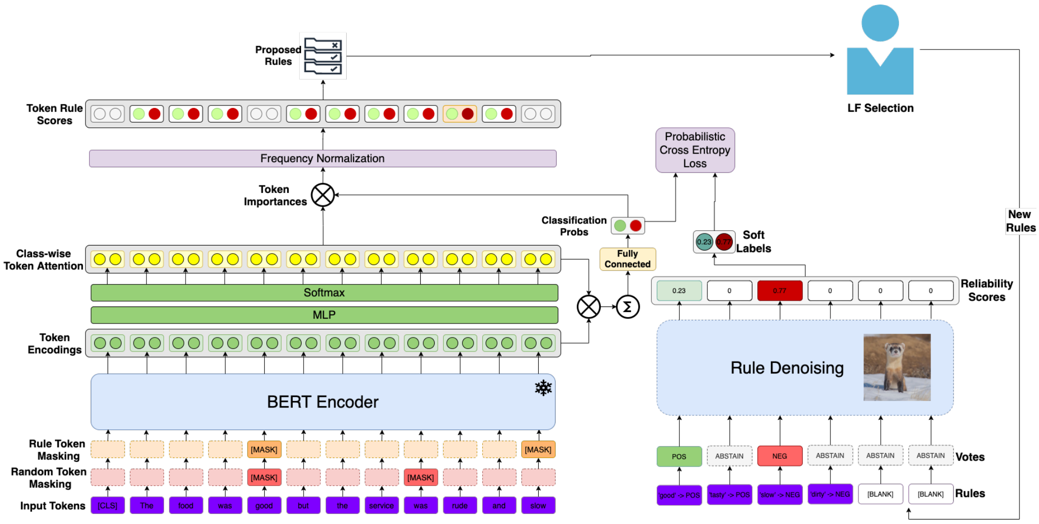


Figure 3. Model architecture for REGAL.

We create TextEncoder as a bidirectional, transformer-based encoder [23] using a pretrained, uncased BERT-base model [3] provided by Huggingface [24]. It allows each token's embedding to be conditioned on all other input text in the document, allowing them to capture rich contextual information. We use the outputs $h_{i,t}$ of the last layer as token embeddings:

$$\left[\mathbf{h}_{i,1}, \ldots, \mathbf{h}_{i,T}\right] = \mathbf{enc}([v_{i,1}, \ldots, v_{i,T}]) \tag{1}$$

We will henceforth let $H_i = \left[\mathbf{h}_{i,1}, \ldots, \mathbf{h}_{i,T}\right]$ represent the sequence token embeddings from document d_i.

In addition to initializing TextEncoder with a BERT-base, we encourage the encoder to further learn contextual information about labeling rules using a masked language modeling (MLM) objective. Our masking budget consists of all of tokens used in LFs as well as a random 10% of tokens from the sequence. Each token is either masked or noised according to the strategy in Devlin et al. [3], and TextEncoder is required to predict the correct token in each case. Thus, TextEncoder continually learns new labeling cues rather than memorizing simple labeling functions. Optimization is performed using cross entropy loss over the masked/noised tokens, denoted as $\mathcal{L}_{MLM}$.

3.2. Snippet Selector

After producing expressive token embeddings, those most useful for creating labeling rules must be selected. Accordingly, we develop a SnippetSelector module to identify which pieces of text are most useful for developing precise labeling functions and rich document representations.

SnippetSelector learns to extract words and phrases that are indicative of an individual class label. A classwise attention mechanism over tokens identifies and extracts the token and document level information necessary to generate expressive, class-specific labeling functions. SnippetSelector also calculates each document's probability of belonging to each class. These probabilities serve as suitability scores as to how well-equipped a document is to generate LF keywords of that class.

SnippetSelector takes as inputs the token embeddings from the document encoder and produces class-specific token attention $a_{i,t}^{(c)}$, document embeddings $\mathbf{e}_i$, and document-level class probabilities $\mathbf{p}_i = \left[p_i^{(1)}, \ldots, p_i^{(C)}\right]$, which are computed as follows.

First, class-specific attention scores are calculated for each token in our document. Class-specific attention scores are used by the rule proposal network to generate new labeling rules and are calculated as follows:

$$\mathbf{a}_{i,t} = W_2^a \tanh\left(W_1^a H_i\right) \tag{2}$$

where $W_1^a \in \mathcal{R}^{m_2 \times m_1}$ and $W_2^a \in \mathcal{R}^{c \times m_2}$. These scores are then used to calculate a class-specific document representation

$$\tilde{\mathbf{e}}_i^{(c)} = \sum_{t=1}^{T} \mathbf{a}_{i,t}^{(c)} \mathbf{h}_{i,t} \tag{3}$$

These are in turn aggregated into an overall document representation with class weights η_c

$$\mathbf{e}_i = \sum_{c=1}^{C} \eta_c \tilde{\mathbf{e}}_i^{(c)} \tag{4}$$

This representation will be used by the rule attention submodule to estimate conditional LF reliability.

The class-specific embeddings $\tilde{\mathbf{e}}_i^{(c)}$ are also used to compute the document's class probabilities:

$$\mathbf{p}_i = \text{softmax}\left(\left[\hat{p}_i^{(1)}, \ldots, \hat{p}_i^{(C)}\right]\right) \tag{5}$$

where $p_i^{(c)} = \mathbf{w}_p^{(c)^T} \tilde{\mathbf{e}}_i^{(c)}$ and $\mathbf{w}_p^{(c)}$ is a weight vector corresponding to each class. In addition to serving as this submodule's prediction of the document's label, these probabilities also serve as measures of the document's suitability to contribute to LFs of each particular class.

Because BERT tokens are wordpiece subword units, the `SnippetSelector` aggregates subword attentions to a word level by simply summing all of the subword attentions that correspond to a particular word. These are further aggregated into phrase weights by summing over all words in a phrase. Phrase attentions are then passed to the rule proposal network to create rules that are displayed to users for adjudication.

3.3. Rule Proposal Network

REGAL's `RuleProposer` module REGAL to measure the quality of keyword and phrase based rules given a set of seed rules. This can be easily extended to create rules from a set of seed labels as well. The `RuleProposer` takes as inputs both the class-conditioned word level attention $\mathbf{a}_{i,t}^c$ and document-level class probabilities $\mathbf{p}_i$ and outputs a score $\tau_j^{(c)}$ for each $v_j \in \mathcal{V}$ corresponding to how strongly v_j represents class c. These scores are calculated as:

$$\tau_j^{(c)} c = \frac{1}{|v_j|^{\gamma}} \sum_{i=1}^{|D|} \sum_{t=1}^{T} \mathbf{1}_{v_{i,t}=v_j} p_i^{(c)} \mathbf{a}_{i,t}^{(c)} \tag{6}$$

Here, $\gamma \in [0,1]$ is a parameter that controls how much REGAL's `RuleProposer` balances between the coverage of a phrase (i.e., how often it occurs) and its instance level importance. Low values of γ favor phrases with high coverage while high values of γ favor LFs based on highly precise phrases with less regard for coverage. Since the types of rules needed may differ as training progresses, we allow users to choose γ for each round of proposed rules. In practice, we find that $\gamma \in [0.5, 0.9]$ tend to produce good rules.

Once candidate rules have been generated, they are passed through a `PhrasePruner` module that filters them to improve coverage and discriminative capacity. The `PhrasePruner` performs two pruning steps. First, it trims rules below a certain count threshold α. Trimming ensures that chosen rules have sufficient coverage to be useful. Second, we perform polarity pruning, which limits candidate phrases to those that a difference of at least β between the first and second highest scoring classes. Polarity pruning ensures that rules are highly specific to a single class and eliminates phrases containing stopwords, punctuation, and other tokens not particularly relevant to distinguishing classes. Scores for all but the highest scoring class are set to 0 to avoid any phrase being chosen as a representative of multiple classes. In practice, we find that $\alpha >= 10$ and $\beta = 0.4/|C|$ tend to work well.

Of the remaining phrase scores $\tau_j^{(c)}$, the `RuleProposer` proposes up to k new LFs for each class by choosing the top k scoring phrases $\{v_1^{(c)}, \ldots, v_k^{(c)}\}$ for each class c'. These tokens each induce a labeling function of the form `HAS(x,` $v_i^{(c)}$`)` $\rightarrow c$, where the class label c is assigned to a text x if it contains the token $v_i^{(c)}$.

3.4. Rule Denoiser

As multiple general-purpose LFs are proposed, it is inevitable that some will conflict. Accordingly, we utilize a rule denoiser developed in [25] to learn probabilistic labels based on the rules matched to each instance.

We train these soft labels and the class probabilities $\mathbf{p}_i$ from `SnippetSelector` using probabilistic cross entropy loss:

$$\mathcal{L}_{TOK} = -\sum_{c=1}^{C} y_i^{(c)} \log(p_i^{(c)}) \tag{7}$$

Note that the methods in this section can easily be modified to support multilabel classification. This could be performed by using multiple label models (one for each class) and by replacing the single multi-class cross entropy loss with sum of the individual binary cross entropy loss terms for each class.

3.5. Model Optimization

The entire model is optimized by minimizing the unweighted sum of the loss functions of its components:

$$\mathcal{L} = \mathcal{L}_{MLM} + \mathcal{L}_{TOK} \tag{8}$$

4. Experiments and Discussion

4.1. Datasets

REGAL's performance is evaluated on a number of sentiment and topic classification datasets:

Yelp is a collection of Yelp restaurant reviews classified according to their sentiment;
IMDB is a set of movie reviews classified according to sentiment [26];
AGnews is a news corpus for topic classification with four classes: sports, technology, politics, and business [27];
BiasBios is a set of biographies of individuals with classes based on profession. We use the binary classification subsets utilized in [28]: (1) Professor vs. Physician, (2) Professor vs. Teacher, (3) Journalist vs. Photographer, and (4) Painter vs. Architect.

Basic summary statistics on our data are found in Table 1.

Table 1. Summary of REGAL data and rule generation parameters. The data below describe the respective sizes of traditional train, validate, and test sets, though REGAL only extracts rules from the train set. Coverage denotes the total coverage of the initial set of seed rules, whereas Bal. Coverage denotes the accuracy after downsampling to balance class-wise labeling propensities.

Dataset	**# Train**	**# Valid**	**# Test**	**# Classes**	**Coverage**	**Bal. Coverage**
Yelp	30,400	3800	3800	2	0.2239	0.1042
IMDB	24,500	500	25,000	2	0.1798	0.1663
AG News	96,000	12,000	12,000	4	0.0963	0.0144
Journalist/Photographer	15,629	500	16,129	2	0.3211	0.2364
Professor/Physician	26,738	500	27,238	2	0.5149	0.3772
Professor/Teacher	11,794	500	12,294	2	0.5195	0.3574
Painter/Architect	5618	500	6118	2	0.4516	0.2650

4.2. Baseline Models

We compare REGAL's ability to identify promising keyword LFs to the baseline models models described below.

4.2.1. Snuba or Reef

Snuba [21], recently renamed Reef, is an automated method of extending the coverage of a small, labeled dataset by automatically generating a subset of labeling functions from this labeled subset. It uses an acquisition function for new LFs consisting of a weighted average of the F1 score on the dev set and the Jaccard distance of a newly proposed rule to the current LF set.

4.2.2. Interactive Weak Supervision

Interactive Weak Supervision [28] is very similar to REGAL and uses active learning to evaluate rules based on the documents they match. IWS evaluates rules via an ensemble of small multi-layer perceptrons (MLPs) and prioritizes the labeling uncertain rules close to the decision boundary using the saddle acquisition function described in [29].

4.2.3. Fully Supervised BERT Model (FS BERT)

A fully-supervised BERT model is used to compare the performance of the labeling models developed from REGAL's proposed rules.

4.3. Training Setup

REGAL requires a user to provide at least some labeling signal to prime the rule generator. Accordingly, we provide three phrase-matching LFs for each class of each dataset. Keywords for seed rules are shown in the Appendix B. If the LF's phrase is found in document d_i, the LF assigns its label; otherwise, it abstains from labeling d_i.

REGAL is run for five rounds of LF extraction with $\alpha = 0.7$ and one epoch of training between each round of rule extraction. Each extracted phrase candidate is required to occur in at least 20 documents to be considered as a labeling function. After each round of training and accumulating rule scores, we take the solicit labels on the top m rules for each class, where $m = min(50, k)$ and k is the number of rules above the polarity threshold. Solicited labels are evaluated by an oracle evaluator which accepts a proposed rule r_j if $accuracy(r_j) > \phi$ on matched samples. We choose $\phi = 0.7$ as our acceptance threshold. Further parameter settings for training can be found in the Appendix C.

4.4. Rule Extraction

REGAL's key feature is the ability to extract expressive, high-coverage labeling rules from text. The ability of REGAL to identify promising rules based on the provided seed rules evaluated.

We compare LFs selected by REGAL to those from other methods based on their coverage and accuracy, each macro-averaged across LFs. We additionally compare how the labeling functions from different models work together to train a label denoising model to generate probabilistic labels of the data. Downstream performance is evaluated using the accuracy and area under the receiver operator characteristic curve (AUC). The results of this comparison are shown in Table 2.

Table 2. Performance comparison of LF extraction methods. LF accuracy and coverage are averaged over all LFs produced by the model. # LFs denotes the total number of LFs selected/predicted by the model, not the number proposed. LM Acc and LM AUC represent the accuracy and area under the ROC curve, respectively, of the probabilistic labels produced by a Snorkel label model. For fully-supervised BERT models (denoted by FS BERT), accuracy and AUC are not computed with a label model. * FS BERT results for AG News taken from [30]. ** For fair comparison with IWS and REEF/Snuba, REGAL and FS BERT macro averages exclude AG News.

Dataset	Model	# LFs	LF Acc	Coverage	LM Acc	LM AUC
AG News	IWS	-	-	-	-	-
	REEF/Snuba	-	-	-	-	-
	REGAL	280	0.912	0.007	0.856	-
	FS BERT *	-	-	-	0.952	-
IMDB	IWS	35	**0.807**	0.065	**0.811**	**0.883**
	REEF/Snuba	50	0.729	**0.068**	0.722	0.787
	REGAL	**193**	0.787	0.017	0.510	0.757
	FS BERT	-	-	-	0.914	0.974
Journalist/ Photographer	IWS	110	0.877	0.033	0.898	**0.958**
	REEF/Snuba	23	**0.894**	**0.142**	**0.910**	0.944
	REGAL	**265**	0.840	0.030	0.733	0.890
	FS BERT	-	-	-	0.954	0.990
Painter/ Architect	IWS	157	0.883	0.032	0.893	0.966
	REEF/Snuba	23	**0.893**	**0.140**	0.874	0.947
	REGAL	**373**	0.876	0.034	**0.897**	**0.977**
	FS BERT	-	-	-	0.968	0.995
Professor/ Physician	IWS	238	0.860	0.042	**0.892**	**0.957**
	REEF/Snuba	26	**0.917**	**0.184**	0.882	0.935
	REGAL	**249**	0.876	0.041	0.794	0.871
	FS BERT	-	-	-	0.951	0.994
Professor/ Teacher	IWS	**218**	0.785	0.030	0.760	**0.928**
	REEF/Snuba	12	0.562	**0.619**	0.782	0.839
	REGAL	211	**0.824**	0.029	**0.813**	0.877
	FS BERT	-	-	-	0.938	0.982
Yelp	IWS	87	0.799	0.047	0.747	0.830
	REEF/Snuba	38	**0.833**	**0.071**	**0.830**	**0.887**
	REGAL	**155**	0.803	0.018	0.770	0.837
	FS BERT	-	-	-	0.960	0.992
macro-average	IWS	140.833	**0.835**	0.041	**0.833**	**0.920**
	REEF/Snuba	28.667	0.805	**0.204**	**0.833**	0.890
	REGAL **	**241**	0.834	0.028	0.753	0.868
	FS BERT **	-	-	-	0.9475	0.988

From these results, we observe that REGAL consistently produces more LFs than other methods, but that the average accuracy of these is often slightly below the LFs produced by Reef and IWS. However, the average accuracy for REGAL could also be distorted if the average accuracy of its rules is lowered by the large number of additional rules not identified by IWS. To examine this, we compared the rules produced by REGAL and IWS using a Mann–Whitney–Wilcoxon [31] test. Specifically, we test the hypothesis that one produces rules that are significantly more accurate than those produced by the other. The results of these tests is given in Table 3. These tests reveal that the accuracy of rules from REGAL and IWS are very comparable, with no significant difference on four of six datasets and each method significantly outperforming the other on one dataset each.

Another interesting result is both models often see lower accuracy from downstream label models than the average accuracy of LFs input into said label models. Upon further investigation, this phenomenon appears to be occur due to imbalance in the total number of labeling votes for each individual class. To test this hypothesis, we balanced the number of noisy label votes to reflect a roughly even class balance. Balancing was performed by randomly downsampling labeling functions from dominant classes until all classes had roughly the same number of total LF votes for each class. The resultant accuracy scores before and after balancing are shown in Table 4. These results reveal that balancing LF outputs tends to increase accuracy for Snorkel label models and for majority voting, despite reducing the amount of data used for training. However, balancing tends to reduce AUC scores, implying that the additional labels do assist in rank-ordering instances even if these instances are mislabeled due to the decision boundary cutoff. Because of this skew, labels and probabilities produced by these label models should be used with care.

Table 3. Statistical comparison of REGAL and IWS using the Mann–Whitney–Wilcoxon (MWW) test. The methods show no significant difference except on the Journalist/Photographer and Professor/Physician datasets. After Bonferroni correction, MWW shows that REGAL outperforms IWS on Professor/Physician and IWS is outperforms REGAL on Journalist/Photographer. * Significant at $p < 0.05$ after Bonferroni correction; ** significant at $p < 0.01$ after Bonferroni correction.

Dataset	Higher Med. Acc.	MWW *p*-val.
Yelp	REGAL	0.3438
IMDB	IWS	0.1926
Journalist/Photographer	**IWS ***	**0.0066 ***
Professor/Teacher	REGAL	0.2086
Professor/Physician	**REGAL ****	**0.0010 ****
Painter/Architect	IWS	0.1438

Table 4. Effects of balancing data on model label model performance. We balanced data by calculating the total number of noisy label votes for each class and randomly replacing votes for dominant classes until all label distribution was approximately balanced. We measure change in total coverage as well as Accuracy and AUC for both Snorkel label models and a simple majority voting LF aggregator (denoted "MV"). Imbalance Ratio reflects the ratio of most labeled class: least labeled class. Note that rows with higher imbalance ratio have tend to see larger improvements in accuracy after balancing.

Dataset	Model	Δ Accuracy	Δ AUC	MV Acc	Δ MV AUC	Δ Coverage	Imbalance Ratio
AG News	**REGAL**	0.011	–	−0.034	–	−0.154	2.245
IMDB	**IWS**	−0.002	−0.014	0.008	0.001	−0.107	1.896
	REEF/Snuba	0.002	0.000	0.000	0.000	−0.002	1.053
	REGAL	0.066	−0.068	0.083	−0.008	−0.165	3.573
Journalist/Photographer	**IWS**	−0.001	−0.013	−0.012	0.001	−0.112	2.492
	REEF/Snuba	−0.003	−0.004	−0.004	0.000	−0.006	1.493
	REGAL	−0.014	0.004	0.025	−0.012	−0.001	1.319
Painter/Architect	**IWS**	0.033	−0.014	0.022	0.007	−0.136	3.969
	REEF/Snuba	0.001	0.000	−0.003	−0.003	−0.004	1.340
	REGAL	−0.011	−0.006	0.015	−0.004	−0.001	1.238
Professor/Physician	**IWS**	−0.010	−0.008	0.006	−0.001	−0.002	1.170
	REEF/Snuba	−0.004	0.001	−0.007	−0.002	0.000	1.499
	REGAL	−0.026	−0.024	0.024	−0.009	0.000	1.380
Professor/Teacher	**IWS**	0.120	−0.033	0.146	0.075	−0.253	7.109
	REEF/Snuba	0.008	0.000	0.000	−0.008	0.000	1.012
	REGAL	−0.001	−0.013	0.000	−0.003	0.000	1.121
Yelp	**IWS**	0.085	0.061	0.060	−0.007	−0.140	3.285
	REEF/Snuba	0.003	0.002	0.001	0.000	−0.008	1.226
	REGAL	0.010	0.012	0.021	−0.019	−0.036	1.642

4.5. Qualitative LF Evaluation

The LFs extracted by REGAL are best understood through specific examples. This enables a user to inspect the extent to which LFs discovered by REGAL model semantically meaningful indicators for a particular domain, or if REGAL is rather targeting artifacts that are specific to the particular dataset in question. To this end, we present the first six rules generated by REGAL for each of our datasets in Table 5. We additionally provide samples of multi-word LFs discovered by REGAL in Table A1 in the Appendix B.

From the top rules selected, we see the type of textual clues REGAL catches to select rules. In Yelp reviews, it unsurprisingly catches words of praise to represent positive reviews and people seeking remediation for poor experiences for negative reviews. Additionally, REGAL selects many specific food entreés as positive LFs keywords, highlighting that positive reviews tend to discuss the individual food that people ordered more than negative ones. In contrast, negative LFs tend to focus on experiences outside of dining, such are retail and lodging.

Table 5. Top 6 unigram labeling functions from first 5 iterations of REGAL. In some cases, REGAL did not identify LFs for particular classes at some iterations, denoted by "-".

Dataset	Class	Iter. 1	Iter. 2	Iter. 3
AG News	**Sports**	‘ioc’, ‘olympic’, ‘knicks’, ‘nba’, ‘ncaa’, ‘medal’	‘mls’, ‘mvp’, ‘fc’, ‘sport’, ‘cowboys’, ‘golf’	‘102’, ‘35th’, ‘vs’, ‘2012’, ‘700th’, ‘ruud’
	Science/Tech	‘microprocessors’, ‘microprocessor’, ‘antivirus’, ‘workstations’, ‘passwords’, ‘mainframe’	‘xp’, ‘os’, ‘x86’, ‘sp2’, ‘worms’, ‘worm’	‘hd’, ‘666666’, ‘src’, ‘sd’, ‘br’, ‘200301151450’
	Politics	‘allawi’, ‘prime’, ‘ayad’, ‘iyad’, ‘kofi’, ‘sadr’	‘plo’, ‘holy’, ‘roh’, ‘troops’, ‘troop’, ‘mp’	-
	Business	‘futures’, ‘indexes’, ‘trading’, ‘investors’, ‘traders’, ‘shares’	‘http’, ‘www’, ‘output’, ‘bp’, ‘dow’, ‘bhp’	‘ob’
IMDB	**Positive**	‘enchanting’, ‘errol’, ‘astaire’, ‘matthau’, ‘witherspoon’, ‘mclaglen’	‘garcia’, ‘ruby’, ‘1939’, ‘emily’, ‘myrna’, ‘poem’	‘delight’, ‘stellar’, ‘vivid’, ‘voight’, ‘burns’, ‘dandy’
	Negative	‘dumbest’, ‘manos’, ‘lame’, ‘whiny’, ‘laughable’, ‘camcorder’	‘pointless’, ‘inept’, ‘inane’, ‘implausible’, ‘abysmal’, ‘cheap’	‘vomit’, ‘joke’, ‘morons’, ‘ugh’, ‘snakes’, ‘avoid’
Journalist/ Photographer	**Photographer**	‘35mm’, ‘shoots’, ‘polaroid’, ‘headshots’, ‘captures’, ‘portraiture’	‘exposures’, ‘kodak’, ‘nudes’, ‘viewer’, ‘imagery’, ‘colors’	‘shadows’, ‘macro’, ‘canvas’, ‘skill’, ‘poses’, ‘hobby’
	Journalist	‘corruption’, ‘government’, ‘cnn’, ‘previously’, ‘policy’, ‘stints’	‘governance’, ‘anchor’, ‘pbs’, ‘npr’, ‘democracy’, ‘bureau’	‘arabic’, ‘programme’, ‘elsewhere’, ‘economy’, ‘crisis’, ‘prior’
Painter/ Architect	**Painter**	‘galleries’, ‘collections’, ‘residencies’, ‘acrylic’, ‘plein’, ‘pastels’	‘impressionist’, ‘textures’, ‘strokes’, ‘flowers’, ‘figurative’, ‘brush’	‘palette’, ‘feelings’, ‘realism’, ‘emotion’, ‘realistic’, ‘filled’
	Architect	‘soa’, ‘enterprise’, ‘bim’, ‘server’, ‘scalable’, ‘solutions’	‘infrastructure’, ‘methodologies’, ‘certifications’, ‘intelligence’, ‘teams’, ‘developer’	‘automation’, ‘computing’, ‘delivery’, ‘healthcare’, ‘initiatives’, ‘processing’
Professor/ Physician	**Professor**	‘banking’, ‘democratization’, ‘verification’, ‘cooperation’, ‘governance’, ‘b’	‘security’, ‘finance’, ‘macroeconomics’, ‘microeconomics’, ‘political’, ‘law’	‘acm’, ‘optimization’, ‘mechanical’, ‘metaphysics’, ‘computational’, ‘visualization’
	Physician	‘specializes’, ‘alaska’, ‘takes’, ‘accepts’, ‘norfolk’, ‘ky’	‘speaks’, ‘aurora’, ‘carolinas’, ‘menorah’, ‘novant’, ‘affiliated’	‘vidant’, ‘anthonys’, ‘southside’, ‘fluent’, ‘hindi’, ‘osf’

Table 5. *Cont.*

Dataset	Class	Iter. 1	Iter. 2	Iter. 3
Professor/ Teacher	**Teacher**	'grades', 'ages', 'eighth', 'aged', 'graders', 'grade'	'ratings', 'sixth', 'fifth', 'fun', 'fourth', 'tutoring'	'pupils', 'favorite', 'cooking', 'volunteering', 'comparing', 'friends'
	Professor	'governance', 'constitutional', 'co-operation', 'regulation', 'democracy', 'finance'	'econometrics', 'banking', 'economy', 'markets', 'entrepreneurship', 'economic'	'globalization', 'optimization', 'firms', 'statistical', 'conflict', 'tax'
Yelp	**Positive**	'phenomenal', 'yummy', 'delectable', 'favorite', 'amazing', 'atmosphere'	'terrific', 'heavenly', 'notch', 'hearty', 'chic', 'stylish'	'handmade', 'kale', 'cozy', 'carpaccio', 'tender', 'fave'
	Negative	'refund', 'pharmacy', 'disrespectful', 'refunded', 'warranty', 'rudest'	'cancel', 'scam', 'confirmed', 'dealership', 'driver', 'appt'	'receipt', 'confirm', 'reply', 'cox', 'clerk', 'policy'

Similar trends emerge in LFs selected for the professor/physician dataset. 'Professor' LFs tend to correspond to academic disciplines, whereas 'physician' LFs relate to aspects of medical practice (such as specialization or insurance) of the specific location where a physician practiced. Notably, the locations selected as rules for the physician class are lesser-known, avoiding towns with major universities that may conflict with the professor class.

Note that all of the rules selected were confirmed by oracle evaluation. This implies that REGAL selects some LFs that are data artifacts that correlate closely with one class but are not intuitive to a human annotator. In this sense, REGAL can be a useful tool for identifying artifacts that could impede the generalization of a model and be used to make models more robust.

5. Related Work

REGAL builds on dual foundations, active and weakly supervised learning, for text classification.

5.1. Active Learning

REGAL shares a few goals with active learning. First, REGAL iteratively solicits user feedback to train a robust downstream model with minimal annotation effort. Methods to perform active learning include selecting a diverse, representative set of instances to annotate [32,33], selecting the instances about which the model is least confident [29,34], and selecting the instances with the highest expected gradient norm and thus the highest expected model change [35]. Second, REGAL shares active learning's goal to interactively solicit an optimal set of labels. However, REGAL differs by soliciting labels for labeling functions rather than individual data points. Soliciting labels for label functions increase coverage for a much larger number of instances per given label. It also enables LFs to be inductively applied to additional data not seen during training.

5.2. Weakly Supervised Learning

Weakly supervised learning dates back to early efforts to model the confidence of crowd-sourced labels based on inter-annotator agreement [13]. Works such as Snorkel [12,25] have adapted these ideas to learn label confidence based on the aggregation of large numbers of noisy, heuristic LFs. Weak supervision has been shown to be effective at a host of tasks, including named entity recognition [36,37], seizure detection [38], image segmentation [39], relation extraction [20], and text classification [9,11,18,40]. However, all of these models require users to define labeling functions manually, creating a usability barrier to subject matter experts not used to writing code. Some also require additional labeled instances for self-training [18,40], which REGAL does not. Recent works have reduced the barrier to scaling weak supervision by propagating labels to nearby matched examples in latent space [41] and soft-matching LFs to samples not explicitly labeled by the LF [20]. Additional studies have shown that convergence to a final set of diverse LFs can be accelerated by prompting users with high-priority examples such as those which are unlabeled or have conflicting LFs [19].

Snuba/Reef [21] uses similar weak supervision. Snuba/Reef generates LFs from a small labeled set of data and iteratively creates a diverse set of by adding new LFs using an acquisition $(r_k) = w * f_{score} + (1 - w) * j_{score}$, where f_{score} is the F1 score of the rule on the labeled dev set, j_{score} is the Jaccard similarity of the rule to the currently labeled set, and $w \in [0, 1]$ a weight parameter. Snuba differs from our method in that it requires labeled data in order to generate labeling functions and it does provide a means of interactive human feedback for LF selection.

5.3. Combined Active Learning with Interactive Weak Supervision

REGAL is the second known work to combine active learning with interactive weak supervision for text classification using LFs. IWS [28] also enables interactive weak supervision via active learning on labeling functions. Similar to REGAL, IWS begins by enumerating all labeling functions from a particular "LF family," such as all of the n-grams in a document. It featurizes LFs using the SVD of their matched documents, then uses an ensemble of small neural networks to estimate the accuracy of each LF. IWS then treats selecting useful LFs as an active level set estimation problem, using the saddle acquisition function Bryan et al. [29]. IWS is similar to REGAL in that both interactively select n-gram LFs via human feedback.

REGAL differs from IWS in two main areas. First, REGAL seeks attention on embeddings from pretrained language models to optimally select quality n-gram LFs, whereas IWS uses an ensemble of weak classifiers to estimate a distribution of LF quality. Second, REGAL uses a different acquisition function than IWS. REGAL seeks to maximize a combination of coverage and accuracy of proposed LFs (i.e., optimizing LF quality), whereas IWS seeks to find LFs near the decision boundary about which it is uncertain.

6. Conclusions and Future Work

REGAL interactively creates high-quality labeling patterns from raw text, enabling an annotator to more quickly and effectively label a data set. REGAL improves upon the challenges of label noise, label incompleteness, and annotator effort. Results confirm the combination of weak supervision with active learning provides strong performance that accelerates advancements in low-resource NLP domains by assisting human subject matter experts in labeling their text data.

Future work to improve REGAL and other interactive weak supervision methods will need to improve rule denoising and LF generation. While REGAL can identify useful labeling rules, these rules often result in unbalanced labels that skew training and overpower denoising methods meant to synthesize them. Better denoising algorithms are needed to be able to deal with this imbalance, which will also improve the performance of models such as REGAL that interact with these probabilistic labels. Given that most label models expect LFs to be fully specified before training, future work that identifies fast ways to update models with the addition of new LFs would be particularly useful. Additional work could also explore ways to generate and extract labeling functions from other, more expressive families such as regular expressions to create more precise LFs or automatically refine existing ones. More expressive labeling functions could also support sequence tagging tasks such as named entity recognition, e.g., in [36].

Author Contributions: Conceptualization, D.K., C.Z. and C.S.M.; methodology, D.K., D.N.A., W.R., C.Z. and C.S.M.; software, D.K. and D.N.A.; formal analysis, D.K.; writing—original draft preparation, D.K. and C.S.M.; writing—review and editing, D.K., D.N.A., W.R., C.Z. and C.S.M.; visualization, D.K.; supervision, C.Z. and C.S.M.; project administration, C.S.M.; funding acquisition, C.S.M. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by National Science Foundation CAREER grant 1944247 to C.M., National Institute of Health grant R21-CA232249 to C.M., Whitaker Biomedical Engineering Seed Grant for COVID-19 to C.M., the McCamish Parkinson's Disease Innovation Program at Georgia Institute of Technology and Emory University to C.M., Alzheimer's Association Research Grant Award 018-AARGD-591014 to C.M., Goizueta Alzheimer's Disease Research Center at Emory University grant awards P50 AG025688 and P30 AG066511 to C.M., Children's Hospital of Atlanta Aflac Pilot Grant Award to C.M., National Institute of Health grant U19-AG056169 sub-award to C.M., National Science Foundation award III-2008334 to C.Z., an Amazon Faculty Award to C.Z., and a Google Faculty Award to C.Z.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Access to download the REGAL code can be found on GitHub: www.github.com/pathology-dynamics/regal; accessed on 17 December 2021.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

Abbreviations

The following abbreviations are used in this manuscript:

LF	Labeling Function
LM	Label Model
AUC	Area Under the Curve
IWS	Interactive Weak Supervision
REGAL	Rule-Enhanced Generative Active Learning

Appendix A. Datasets and Preprocessing

For each of our datasets, we held out small validation and test sets to ensure that REGAL was training properly and to evaluate how created LFs were contributing to model performance. Validation and test sets were not used in rule selection. A summary of the statistics of each of our datasets can be found in Table 1 in the main ppaer. We also include the coverage of our initial seed rules and the size of the balanced, downsampled labeled dataset used to begin model training.

Minimal preprocessing was performed only to preserve consistency in the text. Text in datasets was processed to remove accents and special characters. All examples were truncated or padded to 128 word-piece tokens.

While stopwords and punctuation were not removed from the text, LFs with punctuation were removed, as were unigram LFs that were included as stopwords. For sentiment datasets, we converted contractions containing a negation into their non-contracted form (e.g., "didn't" became "did not") to insure consistency.

Appendix B. Seed Labeling Rules

Here we describe the three seed rules used for each class for each baseline dataset.

Yelp LFs:

Positive: 'best', 'excellent', 'awesome'
Negative: 'worst', 'awful', 'nasty'

Professor/Physician LFs:

Professor: 'professor', 'science', 'published'
Teacher: 'medical', 'practice', 'physician'

Journalist/Photographer LFs:

Journalist: 'journalism', 'writing', 'news'
Photographer: 'photographer', 'studio', 'fashion'

Professor/Teacher LFs:

Professor: 'professor', 'research', 'published'
Teacher: 'elementary', 'children', 'teacher'

Painter/Architect LFs:

Painter: 'painting', 'art', 'gallery'
Architect: 'building', 'architect', 'residential'

IMDB LFs:

Positive: 'masterpiece', 'excellent', 'wonderful'
Negative: 'worst', 'awful', 'garbage'

Table A1. Top 6 bigram labeling functions from the first iteration of REGAL. Cases where REGAL did not select any bigram LFs are denoted by "-".

Dataset	Class	Length 2 Rules
AG News	**Sports**	'2006 world', '93 -', '- star', 'half goals', 'world short', '1 draw'
	Science/Tech	'worm that', 'os x', '/ l', 'data -', 'a flaw', 'chart)'
	Politics	'labour party', 'labor party', 's party', 'al -', 'bush '', 'pro -'
	Business	'- wall', '$ 46', 'up 0', '$ 85', '$ 43', 'a &'
IMDB	**Positive**	'kelly and', 'claire danes', 'george burns', 'jack lemmon', 'michael jackson', 'hong kong'
	Negative	'just really', 'plain stupid', 'maybe if', 'avoid it', 'so stupid', 'stupid the'
Journalist/ Photographer	**Photographer**	-
	Journalist	'see less', 'twitter :'
Painter/ Architect	**Painter**	'attended the', 'public collections', 'collections including'
	Architect	-
Professor/ Physician	**Professor**	'of financial', 'see less', 'film and', 'and society', 'fiction and', '_ b'
	Physician	'oh and', 'va and', 'la and', 'tn and', 'ca and', 'ok and'
Professor/ Teacher	**Teacher**	'childhood education', 'early childhood', 'primary school', 'of 4', 'special education', 'rating at'
	Professor	'modeling and', 'and computational', 'climate change', 'and organizational', 'of government', 'nsf career'
Yelp	**Positive**	'affordable and', 'food good', 'highly recommended', 'highly recommend', 'top notch', 'definitely recommend'
	Negative	'never again', 'never recommend', 'ever again', 'very bad', 'never going', 'my card'

Appendix C. Additional Training Details

We trained REGAL for 5 epochs on the labeled subset of data using a batch size of 128 and Adam optimizer with a learning rate 0.0001. At the end of each epoch, REGAL proposed 40 rules for each class. After each epoch, we reset model weights as done in traditional active learning [8] to prevent overfitting and corresponding rule quality degradation. We additionally stopped REGAL early if, after an epoch of training, no rules with sufficient high class polarity were available to propose. Lack of rules with sufficient high class polarity indicated that performance had saturated.

During training, some label classes tend to more readily yield viable, high-coverage rules than others, which leads to imbalance in the noisy labels. This phenomenon cripples the label denoising model, which impedes model training and the learning of additional rules. We solve this problem by randomly downsampling the noisy labels during model training to contain roughly equal numbers of each class. This leads to greater stability in LF generation.

For all binary datasets, we followed the example of [11] by requiring all labeled examples used in training to be matched by at least 2 LFs during training for greater model stability.

Appendix D. Generated Rules

In addition to the keyword rules displayed in the main text, REGAL is capable of generating n-gram rules of arbitrary length. This allows REGAL to identify more complex linguistic phenomena. N-gram rules (for $n > 1$ generally have substantially lower coverage than unigram rules and may or may not improve downstream model performance. In this section, we display a sample of 2-gram rules discovered for each dataset in Table A1. We additionally allowed our models to search for rules with more tokens, but discovered that n-gram rules for $n \geq 3$ are generally less valuable. However, it is possible longer n-grams may be useful for other types of datasets.

Appendix E. Mann–Whitney-Wilcoxon Test

The Mann–Whitney-Wilcoxon test [31,42] is a non-parametric alternative for comparing independent samples of two non-normally distributed random variables. It tests the null hypothesis that both samples (samples "a" and "b" o) are drawn from the same population P. To perform the test, first pool all samples from a and b together and rank them from 1 to N, where $N = n_a + n_b$ is the combined sample size of the two groups. Next, divide the ranked samples into their respective sub-populations and calculate $T_a = \sum_{r \in a} r$, i.e., the sum of the ranks of samples in a. T_b is calculated similarly. Finally, the uniformly distributed test statistic, U, is calculated as:

$$U = \begin{cases} T_a - \frac{n_a(n_a+1)}{2}, & \text{if } n_a > n_b \\ T_b - \frac{n_b(n_b+1)}{2}, & \text{otherwise} \end{cases} \tag{A1}$$

Values of the Mann–Whitney-Wilcoxon test were computed using the statistics module of the `scipy` python package [43]. A standard Bonferonni correction was utilized to lower the p-value threshold for significance in the case of multiple comparisons. A traditional family-wise alpha equal to 0.05 was used. The Bonferonni correction takes the family-wise alpha and divides it be the number of relevant performed comparisons for a given group of pairwise tests utilizing the same sample(s). The Bonferroni correction for multiple pair-wise comparisons is a conservative assessment of significance, as it decreases the likelihood of a Type 1 error (i.e., a false positive - rejecting the null hypothesis when it is, in fact, true).

References

1. Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. Language Models are Few-Shot Learners. In *Advances in Neural Information Processing Systems*; Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M.F., Lin, H., Eds.; Curran Associates, Inc.: New York, NY, USA, 2020; Volume 33, pp. 1877–1901.
2. Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; Liu, P.J. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *J. Mach. Learn. Res.* **2020**, *21*, 1–67.
3. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Minneapolis, MN, USA, 2–7 June 2019; pp. 4171–4186. [CrossRef]
4. Peters, M.E.; Neumann, M.; Iyyer, M.; Gardner, M.; Clark, C.; Lee, K.; Zettlemoyer, L. Deep contextualized word representations. In Proceedings of the NAACL, New Orleans, LA, USA, 1–6 June 2018.
5. Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; Stoyanov, V. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv* **2019**, arXiv:1907.11692.
6. Lee, J.; Yoon, W.; Kim, S.; Kim, D.; Kim, S.; So, C.H.; Kang, J. BioBERT: A pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics* **2019**, *36*, 1234–1240. [CrossRef] [PubMed]
7. Beltagy, I.; Lo, K.; Cohan, A. SciBERT: Pretrained Language Model for Scientific Text. *arXiv* **2019**, arXiv:1903.10676.
8. Ein-Dor, L.; Halfon, A.; Gera, A.; Shnarch, E.; Dankin, L.; Choshen, L.; Danilevsky, M.; Aharonov, R.; Katz, Y.; Slonim, N. Active Learning for BERT: An Empirical Study. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Online, 16–20 November 2020; pp. 7949–7962.

9. Rühling Cachay, S.; Boecking, B.; Dubrawski, A. End-to-End Weak Supervision. *Adv. Neural Inf. Process. Syst.* **2021**, *34*. Available online: https://proceedings.neurips.cc/paper/2021/hash/0e674a918ebca3f78bfe02e2f387689d-Abstract.html (accessed on 16 December 2021).
10. Yu, Y.; Zuo, S.; Jiang, H.; Ren, W.; Zhao, T.; Zhang, C. Fine-Tuning Pre-trained Language Model with Weak Supervision: A Contrastive-Regularized Self-Training Approach. In Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Mexico City, Mexico, 6–11 June 2021; pp. 1063–1077.
11. Ren, W.; Li, Y.; Su, H.; Kartchner, D.; Mitchell, C.; Zhang, C. Denoising Multi-Source Weak Supervision for Neural Text Classification. In Proceedings of the Findings of the Association for Computational Linguistics: EMNLP 2020, Online, 16–20 November 2020; pp. 739–3754.
12. Ratner, A.; Bach, S.H.; Ehrenberg, H.; Fries, J.; Wu, S.; Ré, C. Snorkel. *Proc. VLDB Endow.* **2017**, *11*, 269–282. [CrossRef] [PubMed]
13. Dawid, A.P.; Skene, A.M. Maximum likelihood estimation of observer error-rates using the EM algorithm. *J. R. Stat. Soc. Ser. C Appl. Stat.* **1979**, *28*, 20–28. [CrossRef]
14. Settles, B. *Active Learning Literature Survey*; 2009.
15. Nodet, P.; Lemaire, V.; Bondu, A.; Cornuéjols, A.; Ouorou, A. From Weakly Supervised Learning to Biquality Learning: An Introduction. In Proceedings of the 2021 International Joint Conference on Neural Networks (IJCNN), Shenzhen, China, 18–22 July 2021; pp. 1–10. [CrossRef]
16. Ratner, A.; Hancock, B.; Dunnmon, J.; Sala, F.; Pandey, S.; Ré, C. Training complex models with multi-task weak supervision. In Proceedings of the AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019; Volume 33, pp. 4763–4771.
17. Meng, Y.; Shen, J.; Zhang, C.; Han, J. Weakly-supervised neural text classification. In Proceedings of the International Conference on Information and Knowledge Management, Proceedings, Turin, Italy, 22–26 October 2018; pp. 983–992. [CrossRef]
18. Awasthi, A.; Ghosh, S.; Goyal, R.; Sarawagi, S. Learning from Rules Generalizing Labeled Exemplars. In Proceedings of the International Conference on Learning Representations, Addis Ababa, Ethiopia, 26–30 April 2020.
19. Cohen-Wang, B.; Mussmann, S.; Ratner, A.; Ré, C. Interactive programmatic labeling for weak supervision. In Proceedings of the KDD DCCL Workshop, Anchorage, AK, USA, 4–8 August 2019.
20. Zhou, W.; Lin, H.; Lin, B.Y.; Wang, Z.; Du, J.; Neves, L.; Ren, X. Nero: A neural rule grounding framework for label-efficient relation extraction. In Proceedings of the Web Conference 2020, Taipei Taiwan, 20–24 April 2020; pp. 2166–2176.
21. Varma, P.; Ré, C. Snuba: Automating weak supervision to label training data. In Proceedings of the VLDB Endowment. International Conference on Very Large Data Bases, Rio de Janeiro, Brazil, 12 August 2018; Volume 12, p. 223.
22. Qu, M.; Ren, X.; Zhang, Y.; Han, J. Weakly-supervised Relation Extraction by Pattern-enhanced Embedding Learning. In Proceedings of the 2018 World Wide Web Conference on World Wide Web—WWW '18, Florence, Italy, 18–22 May 2018; Association for Computing Machinery (ACM): New York, NY, USA, 2018; pp. 1257–1266. [CrossRef]
23. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, A.; Polosukhin, I. Attention is all you need. *Adv. Neural Inf. Process. Syst.* **2017**, *2017*, 5999–6009.
24. Wolf, T.; Debut, L.; Sanh, V.; Chaumond, J.; Delangue, C.; Moi, A.; Cistac, P.; Rault, T.; Louf, R.; Funtowicz, M.; et al. Transformers: State-of-the-Art Natural Language Processing. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, Association for Computational Linguistics, Online, 16–20 November 2020; pp. 38–45.
25. Varma, P.; Sala, F.; He, A.; Ratner, A.; Re, C. Learning Dependency Structures for Weak Supervision Models. In Proceedings of the Machine Learning Research, Vancouver, BC, Canada, 8–14 December 2019; Volume 97, pp. 6418–6427.
26. Maas, A.L.; Daly, R.E.; Pham, P.T.; Huang, D.; Ng, A.Y.; Potts, C. Learning word vectors for sentiment analysis. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies—Volume 1, Portland, OR, USA, 19–24 June 2011; pp. 142–150.
27. Zhang, X.; Zhao, J.; LeCun, Y. Character-level convolutional networks for text classification. In Proceedings of the 29th Annual Conference on Neural Information Processing Systems, NIPS 2015, Montreal, QC, Canada, 7–12 December 2015; pp. 649–657.
28. Boecking, B.; Neiswanger, W.; Xing, E.; Dubrawski, A. Interactive Weak Supervision: Learning Useful Heuristics for Data Labeling. In Proceedings of the International Conference on Learning Representations, Vienna, Austria, 4–8 May 2021.
29. Bryan, B.; Schneider, J.; Nichol, R.; Miller, C.J.; Genovese, C.R.; Wasserman, L. *Active Learning for Identifying Function Threshold Boundaries*; 2005; pp. 163–170. Available online: https://proceedings.neurips.cc/paper/2005/hash/8e930496927757aac0dbd2438cb3f4f6-Abstract.html (accessed on 16 December 2021).
30. Sun, C.; Qiu, X.; Xu, Y.; Huang, X. How to fine-tune bert for text classification? *China National Conference on Chinese Computational Linguistics*; Springer: Kunming, China, 2019; pp. 194–206.
31. Mann, H.B.; Whitney, D.R. On a test of whether one of two random variables is stochastically larger than the other. *Ann. Math. Stat.* **1947**, *18*, 50–60. [CrossRef]
32. Gissin, D.; Shalev-Shwartz, S. Discriminative active learning. *arXiv* **2019**, arXiv:1907.06347.
33. Sener, O.; Savarese, S. Active learning for convolutional neural networks: A core-set approach. *arXiv* **2017**, arXiv:1708.00489 .
34. Lewis, D.D.; Gale, W.A. *A Sequential Algorithm for Training Text Classifiers*; SIGIR'94; Springer: London, UK, 1994; pp. 3–12.
35. Huang, J.; Child, R.; Rao, V.; Liu, H.; Satheesh, S.; Coates, A. Active learning for speech recognition: The power of gradients. *arXiv* **2016**, arXiv:1612.03226.
36. Li, Y.; Shetty, P.; Liu, L.; Zhang, C.; Song, L. BERTifying the Hidden Markov Model for Multi-Source Weakly Supervised Named Entity Recognition. *arXiv* **2021**, arXiv:2105.12848.

37. Lison, P.; Hubin, A.; Barnes, J.; Touileb, S. Named entity recognition without labelled data: A weak supervision approach. *arXiv* **2020**, arXiv:2004.14723.
38. Saab, K.; Dunnmon, J.; Ré, C.; Rubin, D.; Lee-Messer, C. Weak supervision as an efficient approach for automated seizure detection in electroencephalography. *NPJ Digit. Med.* **2020**, *3*, 1–12. [CrossRef] [PubMed]
39. Hooper, S.; Wornow, M.; Seah, Y.H.; Kellman, P.; Xue, H.; Sala, F.; Langlotz, C.; Re, C. Cut out the annotator, keep the cutout: better segmentation with weak supervision. In Proceedings of the International Conference on Learning Representations, Addis Ababa, Ethiopia, 26–30 April 2020.
40. Karamanolakis, G.; Mukherjee, S.S.; Zheng, G.; Awadallah, A.H. Self-Training with Weak Supervision. In Proceedings of the 2021 Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT 2021), Mexico City, Mexico, 6–11 June 2021.
41. Chen, M.F.; Fu, D.Y.; Sala, F.; Wu, S.; Mullapudi, R.T.; Poms, F.; Fatahalian, K.; Ré, C. Train and You'll Miss It: Interactive Model Iteration with Weak Supervision and Pre-Trained Embeddings. *arXiv* **2020**, arXiv:2006.15168.
42. Wilcoxon, F. Individual comparisons by ranking methods. In *Breakthroughs in Statistics*; Springer: New York, NY, USA, 1992; pp. 196–202.
43. Virtanen, P.; Gommers, R.; Oliphant, T.E.; Haberland, M.; Reddy, T.; Cournapeau, D.; Burovski, E.; Peterson, P.; Weckesser, W.; Bright, J.; et al. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nat. Methods* **2020**, *17*, 261–272. [CrossRef] [PubMed]