# Exploring Inspiration Sets in a Data Programming Pipeline for Product Moderation

**Justine Winkler[1], Simon Brugman[1], Bas van Berkel[2], Martha Larson[1]**
[1]Radboud University, Netherlands
[2]bol.com, Netherlands
m.larson@cs.ru.nl

## Abstract

We carry out a case study on the use of data programming to create data to train classifiers used for product moderation on a large e-commerce platform. Data programming is a recently-introduced technique that uses human-defined rules to generate training data sets without tedious item-by-item hand labeling. Our study investigates methods for allowing product moderators to quickly modify the rules given their knowledge of the domain and, especially, of textual item descriptions. Our results show promise that moderators can use this approach to steer the training data, making possible fast and close control of classifiers that detect policy violations.

## 1 Introduction

Text classifiers play an important role in filtering inappropriate products on e-commerce platforms. Product moderators are dependent on classifiers that have been trained on up-to-date labeled data in order to keep pace with policy changes and new instances of inappropriate products. For example, Amazon had to take fast action to remove offensive T-shirts during the 2020 US election (Bryant, 2020) and overpriced items and fake cures during the COVID-19 pandemic (BBC, 2020). In this paper, we carry out a case study at a large e-commerce platform. We investigate an approach that allows moderators to rapidly steer the creation of labeled training data, thereby enabling close control of moderation classifiers.

Our approach makes use of a recently-introduced technique called *data programming* (Ratner et al., 2016), which generates classifier training data on the basis of rules that have been specified by domain experts (platform moderators). Data programming eliminates the need to individually hand-label training data points. We propose a feedback loop that selects subsets of data, called *inspiration sets*, that are used by moderators as the basis for updating an initial or existing set of rules. We investigate whether inspiration sets can be selected in an unsupervised manner, i.e., without ground truth.

The contribution of our case study is insight into how to support moderators in updating the rules used by a data programming pipeline in a real-world use scenario requiring fast control (i.e., imposing time constraints). Our study is carried out in collaboration with professional moderators at bol.com, a large European e-commerce company. In contrast to our work, most papers on product moderation, such as Arnold et al. (2016), do not obviously take an inside perspective. Most previous studies of data programming, such as Ehrenberg et al. (2016), have looked at user control, but not at *fast* control, i.e., the ability to update rules quickly in order to steer the training data.

Because of the sensitive nature of the work of the platform moderators, our case study is written with a relatively high level of abstraction. We cannot reveal the exact statistics of inappropriate items on the platform. The rules formulated by the moderators are largely based on keywords occurring in the text of product descriptions, but it is not possible to state them exactly. Nonetheless, we find that we are able to report enough information to reveal the potential of inspiration sets for fast control of inappropriate products on e-commerce platforms. This paper is based on a collaborative project with bol.com. Further analysis and experimental results are available in the resulting thesis (Winkler, 2020).

## 2 Related Work

Most work on product moderation (Martin et al., 2018; Xu et al., 2019; Mackey and Kalyanam, 2017) focuses on products sold on social media. In contrast, we study an e-commerce platform from the inside. Like social media moderation, we face the challenge of lexical variation of keywords, cf. Chancellor et al. (2016).

Our study is related to work investigating applications of data programming to a specific problem.



*Proceedings of the 4th Workshop on e-Commerce and NLP (ECNLP 4)*, pages 132–139

Such work includes examples from the medical domain (Callahan et al., 2019; Dutta and Saha, 2019; Dutta et al., 2020; Fries et al., 2019; Saab et al., 2019, 2020), multi-task learning (Ratner et al., 2018, 2019a,b), information extraction (Ehrenberg et al., 2016), and learning discourse structure (Badene et al., 2019). Like our work, such work often adjusts the Snorkel framework (Ratner et al., 2017) for the task at hand.

Previous work has proposed a variety of methods for giving users (who are in our case the product moderators) control over classifiers by making use of a pipeline that allows them to provide feedback about training data labels and classification results. In WeSAL (Nashaat et al., 2018, 2020) user feedback improves the labels that sets of rules assign to data points. In contrast, our focus is on feedback that allows moderators to improve the rules directly. In this respect, our work is related to DDLite (Ehrenberg et al., 2016), which was, to our knowledge, the first to discuss how rules in a data programming pipeline can be improved using sampled data as feedback. Socratic Learning (Varma et al., 2017a,b) considered the issue of users implicitly focusing on subsets of data when they formulate rules, limiting the ability of the data programming pipeline to generalize to data outside of these subsets.

We are working under time-constrained conditions. There are two constraints. First, our moderators are given a limited amount of time to formulate the initial rules. They formulate the rules themselves based solely on their domain expertise and experience, which allows them to work quickly. In contrast, in work such as Ehrenberg et al. (2016) and Ratner et al. (2018), users consult labeled data to formulate the initial rules. Second, our moderators have limited time to *revise* the initial rules. In this step, they consult data in the form of inspiration sets. Wu et al. (2018) investigate time constraints, but focuses on supervised feedback, whereas we also investigate unsupervised approaches.

We consider the work of Cohen-Wang et al. (2019) to be the existing work closest to ours. This work investigates intelligent ways of sampling data points for rule improvement. Our inspiration sets are based on these strategies. A key difference is that Cohen-Wang et al. (2019) simulate their human experts and we work with real domain experts.

| **category** | **train set** | **dev set** | **test set** |
|---|---|---|---|
| fur | 7633 | 406 (69) | 760 (113) |
| illegal wildlife | 7426 | 312 (9) | 627 (20) |
| magnetic balls | 2316 | 340 (5) | 688 (10) |
| weapon knives | 1266 | 210 (17) | 421 (28) |
| smoking-drug | 1071 | 172 (10) | 342 (21) |
| 1-use plastic | 7364 | 454 (124) | 931 (250) |

Table 1: Number of data points in our data sets. For sets with ground truth, the number of points with the positive label, i.e., "inappropriate", is in parentheses.

## 3 Approach

In this section, we describe the data programming pipeline and also our experiment with *inspiration sets*, which investigates the potential for fast control of training data for moderation classifiers.

### 3.1 Policy-based Monitoring Categories

The platform policy of the company we study has five dimensions. It excludes products (1) that are illegal (2) whose production or consumption causes harm (3) that do not match customer expectations (4) that technically fall outside of what the platform can handle (5) that project hate or discrimination. Each dimension contains concrete categories. For example, under (2) there is a category ("single-use plastic"), which contains single-use plastic cups, straws, and cotton swabs that are excluded based on European guidelines. Each of the categories is monitored independently using a classifier, which must detect not only the re-occurring items, but also novel items that are in violation of the platform policy. In this work, we select six typical categories to study: *fur*, *illegal wildlife related*, *magnetic balls* (small enough to be swallowed by children), *weapon-grade knives*, *smoking-drug-related*, and *single-use plastic*.

### 3.2 Data Programming

Figure 1 shows our data programming pipeline. When moderating a product category, product moderators first carry out a "scope" step that identifies the products related to that category (cf. *scoping query*). Then, they carry out a "scan" step that identifies products that violate the policy. The goal of our study is to investigate the usefulness of this pipeline for quickly generating training data to train a classifier that will support the product moderators in carrying out the "scan" step, with a focus on understanding the potential of inspiration sets.

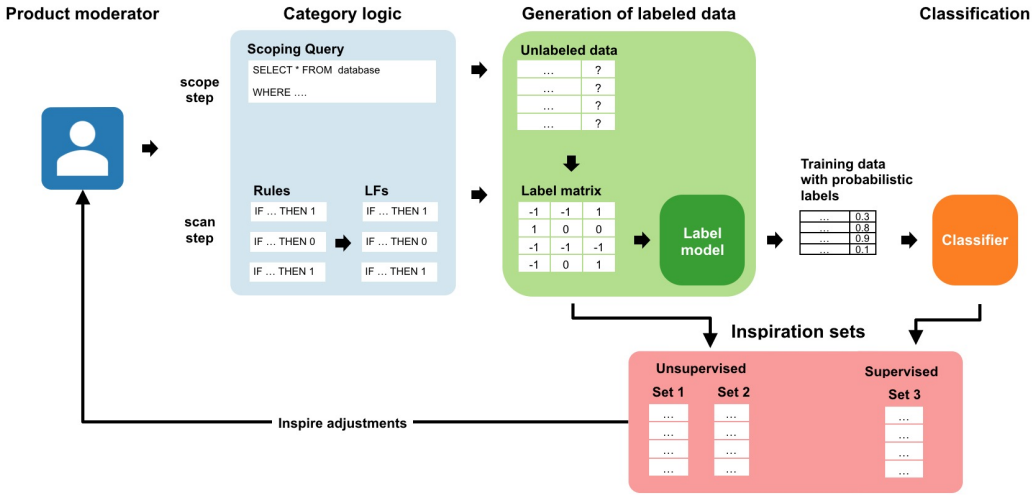


Figure 1: Top row: Our data programming pipeline. Bottom row (red box): Inspiration sets used for fast control.

Data programming (Ratner et al., 2016) is a method that leverages multiple weak supervision signals provided by people who are experts in a domain. The signals take the form of rules, expressed in the form of *labeling functions* (LFs). Given a training data point, an LF either returns a suggested label (0 for "appropriate" or 1 for "inappropriate") or abstains, meaning that it assigns no label. In our study, LFs involve the content of product metadata and keywords in the textual descriptions of products, e.g., `|IF brand == 'brand123' THEN inappropriate ELSE abstain|`. In practice most LFs return only (0, abstain) or (1, abstain). The LFs are applied to the data that was selected in the "scope" step (cf. "Unlabeled data" in Figure 1) to generate a label matrix in which each data point may have multiple, contradictory labels.

In our study, moderators were asked to create rules based on their knowledge of the product categories and their moderation experience. Note that the same moderator was responsible for one category throughout our experiment. They had a limited amount of time (60 min. per category). The time limits in our study were determined in consultation with bol.com's product quality team to simulate real-world settings. This led to an initial set of LFs for each category (number of LFs per category: fur 14, illegal wildlife related 6, magnetic balls 5, weapon-grade knives 5, smoking-drug-related 15, single-use plastic 13).

The label matrix created by the rules is then transformed into labeled data. Ratner et al. (2016) demonstrate that provided a fixed number LFs, a probabilistic labeling model is able to recover a set of labels and corresponding probabilities that can be used to train a classifier (cf. "Training data" and "Classifier" in Figure 1). Snorkel (Ratner et al., 2017) is the first end-to-end system that applies the data programming paradigm. Our case study builds on Snorkel. (More technical details of our setup are in Appendix A.)

### 3.3 Inspiration Sets

We test three different ways of sampling data points to create the inspiration sets consisting of products (cf. Figure 1, bottom). These sets are shown to the moderators to allow them to revise the rules.

**Set 1: Abstain-based strategy** Randomly drawn from training data not yet covered by an LF.

**Set 2: Disagreement-based strategy** Randomly drawn from training data on which LFs disagreed.

**Set 3: Classifier-based strategy** Development data points with largest classifier error.

Set 1 and Set 2 are loosely based on strategies introduced by Cohen-Wang et al. (2019). These strategies are particularly interesting for a real-world setting because they are *unsupervised*, meaning that they are based on information included in the label matrix and do not require ground truth or classifier training. Set 3 is a *supervised* set. It provides product moderators with information about errors that are made by classifiers. This strategy is touched upon, but not implemented, by Cohen-Wang et al. (2019). Recall that Cohen-Wang et al. (2019) uses a simulated human expert, whereas in our experiment, human domain experts inspect the inspiration sets and revise the rules. We used a sim-

| | fur | illegal wildlife | magnetic balls | knives | smoking-drug related | single-use plastic |
|---|---|---|---|---|---|---|
| initial | **0.80** | 0.02 | 0.08 | 0.03 | 0.31 | 0.57 |
| Set 1 | 0.78 | 0.19 | 0.49 | 0.03 | 0.31 | **0.64** |
| Set 2 | 0.78 | 0.00 | **0.65** | 0.03 | **0.36** | 0.57 |
| Set 3 | 0.77 | **0.24** | 0.59 | **0.17** | 0.23 | 0.57 |

Table 2: Data quality results: Label model performance ($F_2$ measure) on the test set.

ple logistic regression classifier for the supervision of Set 3 (see Appendix A.3 for more details).

Each inspiration set contains the number of data points available, up to a maximum of 100. The moderators had a limited amount of time (30 min. per set) to inspect the inspiration sets and add, remove, or change rules in their initial set of rules. Note that in our setting, each inspiration set was drawn once and not updated after the moderator changed one rule.

## 4 Results and Discussion

We analyze how the inspiration sets impact the quality of our data. Table 1 summarizes the data that we use. The ground truth was created by our domain experts. Table 2 presents our results in terms of data quality. Results are reported using the $F_2$ measure due to the importance of recall in our use case. Data points whose "inappropriate" label is generated as having a probability $> 0.5$ are considered positive. Note that scores in Table 2 do not directly reflect the ultimate performance of the classifier, which to a certain extent can leverage data with low $F_2$ scores.

Our results suggest two findings that have, to our knowledge, not been previously documented. First, professional content moderators do not necessarily need labeled sample data to write rules for a data programming pipeline, but instead come quite far relying only on domain knowledge and experience (cf. "initial" in Table 2). Second, when revising their initial set of rules, moderators do not necessarily need an inspiration set created using supervision. Instead, a 30-min. session with an unsupervised inspiration set (Set 1 or Set 2) can improve data quality. The exception is *fur* where $F_2$ is already 0.8, and inspiration sets make the data slightly worse. The category *knives* starts out with extremely low quality data, and inspiration sets do not help much, except for a small, but expensive boost by Set 3, our supervised set. The moderator had only basic experience with this category.

We also found that for most categories, a considerable amount of training data (31-56%) received only abstains (see Appendix B for more details). This observation is consistent with previous work, e.g., that of Cohen-Wang et al. (2019), which has noted that LF sets rarely reach complete coverage. In general, a small number of rules tend to cover a large portion of the data.

The majority of rules had a low precision, and a small number of rules had high recall. Possible reasons are that product moderators tried not to miss out on inappropriate products, or that they had set of specific data points in mind during LF definition, as suggested by Varma et al. (2017a). We also noticed that moderators added and changed, but did not delete rules. In fact, we only observed a single case of a rule being deleted. More research is necessary to understand if this reflects high confidence in the initial choices, or a default thinking pattern, as studied by Adams et al. (2021). Finally, we observe it is important not to assume that each newly added rule yields improvement: rule interactions are also important. A more detailed analysis of the changes brought about by the inspiration sets for two representative cases is included in Appendix C.

## 5 Conclusion and Outlook

Our case study has shown our data programming pipeline can generate labeled data for moderation classifiers in a fraction of the time needed for hand labeling (90 min. vs. a week or more of effort). We have seen that moderators can create effective rules based on their domain knowledge and experience, plus a short exposure to an unsupervised inspiration set. Labeling data by hand in order to create supervised inspiration sets may not be worth the effort. Our observations suggest that it is important that moderators not only write rules, but also continue moderating so that they can gain expertise and also be able to update rules quickly in response to changes in the domain, i.e., a new type of offensive clothing items, as in Bryant (2020).

We hope that our work will inspire research on data programming in domains in which fast response to inappropriate products or content is needed. Future research could seek to understand the ability of moderators to predict the interaction of rules and why they seem hesitant to discard rules once they have created them.

## References


Gabrielle S. Adams, Benjamin A. Converse, Andrew H. Hales, and Leidy E. Klotz. 2021. People systematically overlook subtractive changes. *Nature*, 592(7853):258–261.

Patrick Arnold, Christian Wartner, and Erhard Rahm. 2016. Semi-automatic identification of counterfeit offers in online shopping platforms. *Journal of Internet Commerce*, 15(1):59–75.

Sonia Badene, Kate Thompson, Jean-Pierre Lorré, and Nicholas Asher. 2019. Data programming for learning discourse structure. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 640–645.

BBC. 2020. Coronavirus: Amazon removes overpriced goods and fake cures. 28 February 2020 (Accessed 6 May 2021).

Miranda Bryant. 2020. Amazon removes shirts with derogatory slogan about Kamala Harris. *The Guardian*. 19 Aug 2020 (Accessed 6 May 2021).

Alison Callahan, Jason A. Fries, Christopher Ré, James I. Huddleston, Nicholas J. Giori, Scott L. Delp, and Nigam Haresh Shah. 2019. Medical device surveillance with electronic health records. *npj Digital Medicine*, 2(94).

Stevie Chancellor, Jessica Annette Pater, Trustin Clear, Eric Gilbert, and Munmun De Choudhury. 2016. #thyghgapp: Instagram content moderation and lexical variation in pro-eating disorder communities. In *Proceedings of the 19th ACM Conference on Computer-Supported Cooperative Work & Social Computing*, pages 1201–1213.

Benjamin Cohen-Wang, Stephen Mussmann, Alexander Ratner, and Christopher Ré. 2019. Interactive programmatic labeling for weak supervision. In *Workshop on Data Collection, Curation, and Labeling for Mining and Learning*.

Pratik Dutta and Sriparna Saha. 2019. A weak supervision technique with a generative model for improved gene clustering. In *2019 IEEE Congress on Evolutionary Computation*, pages 2521–2528.

Pratik Dutta, Sriparna Saha, Sanket Pai, and Aviral Kumar. 2020. A protein interaction information-based generative model for enhancing gene clustering. *Scientific Reports*, 10(665).

Henry R. Ehrenberg, Jaeho Shin, Alexander Ratner, Jason A. Fries, and Christopher Ré. 2016. Data programming with DDLite: Putting humans in a different part of the loop. In *Proceedings of the Workshop on Human-In-the-Loop Data Analytics*.

Jason A. Fries, Paroma Varma, Vincent S. Chen, Ke Xiao, Heliodoro Tejeda, Priyanka Saha, Jared Dunnmon, Henry Chubb, Shiraz Maskatia, Madalina Fiterau, et al. 2019. Weakly supervised classification of aortic valve malformations using unlabeled cardiac MRI sequences. *Nature Communications*, 10(3111).

Tim K. Mackey and Janani Kalyanam. 2017. Detection of illicit online sales of fentanyls via Twitter. *F1000Research*, 6:1937.

Rowan O. Martin, Cristiana Senni, and Neil C. D'Cruze. 2018. Trade in wild-sourced African grey parrots: Insights via social media. *Global Ecology and Conservation*, 15:e00429.

Mona Nashaat, Aindrila Ghosh, James Miller, and Shaikh Quader. 2020. WeSAL: Applying active supervision to find high-quality labels at industrial scale. In *Proceedings of the 53rd Hawaii International Conference on System Sciences*, pages 219–228.

Mona Nashaat, Aindrila Ghosh, James Miller, Shaikh Quader, Chad Marston, and Jean-Francois Puget. 2018. Hybridization of active learning and data programming for labeling large industrial datasets. In *2018 IEEE International Conference on Big Data*, pages 46–55.

Alexander Ratner, Stephen H. Bach, Henry Ehrenberg, Jason Fries, Sen Wu, and Christopher Ré. 2017. Snorkel: Rapid training data creation with weak supervision. In *Proceedings of the Very Large Data Bases Endowment*, volume 11, pages 269–282.

Alexander Ratner, Christopher M. De Sa, Sen Wu, Daniel Selsam, and Christopher Ré. 2016. Data programming: Creating large training sets, quickly. In *Advances in Neural Information Processing Systems*, volume 29, pages 3567–3575.

Alexander Ratner, Braden Hancock, Jared Dunnmon, Roger Goldman, and Christopher Ré. 2018. Snorkel metal: Weak supervision for multi-task learning. In *Proceedings of the Second Workshop on Data Management for End-To-End Machine Learning*.

Alexander Ratner, Braden Hancock, Jared Dunnmon, Frederic Sala, Shreyash Pandey, and Christopher Ré. 2019a. Training complex models with multi-task weak supervision. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 4763–4771.

Alexander Ratner, Braden Hancock, and Christopher Ré. 2019b. The role of massively multi-task and weak supervision in software 2.0. In *Proceedings of the Conference on Innovative Data Systems Research*.

Khaled Saab, Jared Dunnmon, Roger Goldman, Alexander Ratner, Hersh Sagreiya, Christopher Ré, and Daniel Rubin. 2019. Doubly weak supervision of deep learning models for Head CT. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 811–819.

Khaled Kamal Saab, Jared Dunnmon, Christopher Ré, Daniel L. Rubin, and Christopher Lee-Messer. 2020. Weak supervision as an efficient approach for automated seizure detection in electroencephalography. *npj Digital Medicine*, 3(59).

Paroma Varma, Bryan He, Dan Iter, Peng Xu, Rose Yu, Christopher De Sa, and Christopher Ré. 2017a. Socratic learning: Correcting misspecified generative models using discriminative models. *arXiv preprint arXiv:1610.08123*.

Paroma Varma, Dan Iter, Christopher De Sa, and Christopher Ré. 2017b. Flipper: A systematic approach to debugging training sets. In *Proceedings of the 2nd Workshop on Human-In-the-Loop Data Analytics*.

Justine Winkler. 2020. Snorkeling for beginners: Applying data programming to product moderation in e-commerce. Master's thesis, Radboud University, Nijmegen, Netherlands.

Sen Wu, Luke Hsiao, Xiao Cheng, Braden Hancock, Theodoros Rekatsinas, Philip Levis, and Christopher Ré. 2018. Fonduer: Knowledge base construction from richly formatted data. In *Proceedings of the 2018 International Conference on Management of Data*, pages 1301–1316.

Qing Xu, Jiawei Li, Mingxiang Cai, and Tim K. Mackey. 2019. Use of machine learning to detect wildlife product promotion and sales on Twitter. *Frontiers Big Data*, 2:28.

# A Technical details of our setup

## A.1 Snorkel

The technical details of the setup we used are as follows: we make use of the official implementation of the Snorkel system. This implementation consolidates work from various publications (Ratner et al., 2017, 2019a) even though the repository name is "snorkel". We used version 0.9.0[1]. There, the label model is optimized using Stochastic Gradient Descent (SGD) on the matrix-completion formulation as in (Ratner et al., 2019a) as opposed to interleaving SGD and Gibbs sampling in (Ratner et al., 2017). In general in data programming, the label model needs two inputs: the dependency structure of the LFs and the class balance of the dependent variable (i.e. $p(Y)$). By default, this implementations assumes the LFs to be conditionally independent and that the class balance is uniformly distributed.

## A.2 Gold labels

For each category of inappropriate items, the product moderator that was specialized in that category labeled the development, validation and test data.

## A.3 Classifier

For each category of inappropriate items, we trained a binary classifier. In line with the official Snorkel introduction tutorial[2], we utilized a simple Logistic Regression classifier. We used categorical cross-entropy loss and an Adam optimizer with a learning rate of 0.01. Note that in this work, we use the classifier for selecting the items in the inspiration Set 3. More details on the whole pipeline can be found in (Winkler, 2020).

# B Properties of the label matrix

In our experiments, inspiration sets inspired the product moderators to adjust their initial set of rules. We translated these rules into LFs in Python. Figure 2 illustrates the impact of the changes to the LFs across all categories of inappropriate items. The leftmost bar of each group represents the coverage of the initial LF sets.

In general, we notice that inspiration sets have an impact on the coverage of the LFs, but that they fall far short from allowing us to achieve full coverage. We also notice, however, that there is a general trend towards inspiration sets increasing the coverage, reflected by a decrease in the fraction of the data set that is assigned 0 labels. This happened in most categories with Set 1 and Set 3 and in half of the categories with Set 2. The strongest coverage increase happened using Set 1.

After the adjustments, for most categories, the LFs within each set seemed to be more coordinated with respect to the data points that they labeled. This can be seen in the increase in the percentage of each data set with multiple labels per sample. However, note that overall, most data points that received a label, received a label from only one LF.

[1] https://github.com/snorkel-team/snorkel/releases/tag/v0.9.0

[2] https://github.com/snorkel-team/snorkel-tutorials/blob/93fc77718b608c5709d4eb8b90b7de7683ba4c15/spam/01_spam_tutorial.ipynb

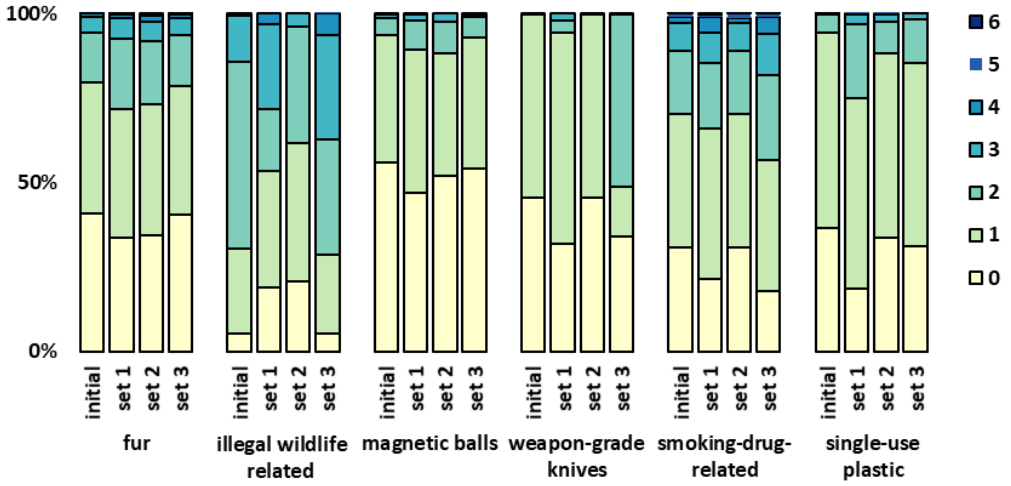


Figure 2: This figure shows the sizes of training data set fractions that received a certain number of labels per sample. Results are shown for the all versions (initial or adjusted using an inspiration set: Set 1, Set 2 or Set 3) of each monitor.

| LF Index | Change | Polarity | Coverage | Overlaps | Conflicts | % gain in $F_2$ |
|---|---|---|---|---|---|---|
| 0 | A | [1] | 0.01 | 0.00 | 0.00 | 18.37 |
| 1 | A | [0] | 0.04 | 0.02 | 0.00 | 1.96 |
| 2 | A | [0] | 0.32 | 0.08 | 0.03 | 16.67 |
| 3 | / | [1] | 0.08 | 0.04 | 0.03 | -21.21 |
| 4 | A | [0] | 0.08 | 0.03 | 0.01 | 0.00 |
| 5 | A | [0] | 0.03 | 0.02 | 0.01 | 0.00 |
| 6 | N | [0] | 0.11 | 0.04 | 0.01 | 1.96 |
| 7 | N | [1] | 0.01 | 0.00 | 0.00 | 0.00 |

Table 3: This table contains the characteristics of the individual LFs for magnetic balls after they have been adjusted with the inspiration Set 1.

## C Individual rule characteristics

In the main paper, we mentioned several observations we made regarding the sets of rules that were created by the professional moderators.

- A small number of rules tend to cover a large portion of the data.
- Moderators added and changed, but did not delete rules (except one rule upon one occasion).
- We cannot not assume that each newly added rule yields improvement.

We based these observations on characteristics that we computed on the training and validation sets in each category. The statistics of these training and validation sets are provided in Table 5.

After translating the rules into LFs, we computed the following characteristics:

- **LF index**: a running index of each rule (Labeling Function) in the set.
- **Change** indicates whether the rules were adjusted (A), newly added (N) or not changed (/) as a result of considering the inspiration set.
- **Polarity**: the polarity that the rule assigns to the training set data points. If the value is [0], then the rule either assigned "appropriate" or abstained. If the value is [1], then the rule either assigned "inappropriate" or abstained. If the value is  then the rule always abstained.
- **Coverage**: the fraction of the training set data points to which the LF assigned a label (i.e., did not abstain).
- **Overlaps**: the fraction of the training set on which the rule assigned a label and at least one other rule did as well (i.e., the rule and at least one other rule did not abstain).
- **Conflicts**: the fraction of the training set on which the labels suggested by multiple rules disagree.

| LF Index | Change | Polarity | Coverage | Overlaps | Conflicts | % gain in $F_2$ |
|---|---|---|---|---|---|---|
| 0 | / | [1] | 0.02 | 0.01 | 0.01 | 4.35 |
| 1 | / | [0] | 0.06 | 0.04 | 0.01 | 0.00 |
| 2 | / | | 0.00 | 0.00 | 0.00 | 0.00 |
| 3 | / | | 0.00 | 0.00 | 0.00 | 0.00 |
| 4 | / | | 0.00 | 0.00 | 0.00 | 0.00 |
| 5 | A | [0] | 0.69 | 0.23 | 0.05 | 0.53 |
| 6 | A | [0] | 0.14 | 0.13 | 0.00 | -0.41 |
| 7 | A | [0] | 0.01 | 0.01 | 0.00 | 0.00 |
| 8 | / | [0] | 0.01 | 0.01 | 0.00 | 0.00 |
| 9 | / | [0] | 0.01 | 0.00 | 0.00 | 0.00 |
| 10 | / | [1] | 0.01 | 0.01 | 0.01 | -0.36 |
| 11 | / | [0] | 0.06 | 0.04 | 0.00 | 0.00 |
| 12 | / | [0] | 0.00 | 0.00 | 0.00 | 0.00 |
| 13 | A | [1] | 0.11 | 0.06 | 0.05 | 73.21 |
| 14 | N | [0] | 0.00 | 0.00 | 0.00 | 0.00 |

Table 4: This table contains the characteristics of the individual LFs for single-use plastic after they have been adjusted with the inspiration Set 1.

| **category** | **training set** | **validation set** |
|---|---|---|
| fur | 7633 | 400 (55) |
| illegal wildlife related | 7426 | 318 (10) |
| magnetic balls | 2316 | 324 (7) |
| weapon-grade knives | 1266 | 210 (18) |
| smoking-drug-related | 1071 | 173 (12) |
| single-use plastic | 7364 | 445 (118) |

Table 5: Number of data points in our training and validation sets. These were the data sets on which we computed the LF characteristics. For convenience, we repeat the sizes of the training data here. Note that the validation sets are disjoint from the development and test sets used in the main paper. For these validation sets, the number of points with the positive label, i.e., "inappropriate", is in parentheses.

- **% gain in $F_2$**: the relative improvement in the $F_2$ score of the labeled data generated by the label model contributed by the individual rule.

Note that Polarity, Coverage, Rules, and Overlap are all calculated on the training data set, and "% gain in $F_2$" is calculated on the validation set.

We chose two representative categories that show the variation of the gain, and provide example analyses for each. The category magnetic balls is in Table 3 and single-use plastic is in Table 4. The analysis uses the rules adjusted after consulting the inspiration Set 1.