Adaptive Rule Discovery for Labeling Text Data

SIGMOD '21

Author
Sainyam Galhotra*
University of Massachusetts Amherst
sainyam@cs.umass.edu

Author
Behzad Golshan
Megagon Labs
behzad@megagon.ai

Author
Wang-Chiew Tan*
Facebook AI
wangchiew@fb.com

Reviewer
鄧雅文
電機工程學系碩士班碩一
R12921059

1. ABSTRACT

Creating and collecting labeled data is one of the major bottlenecks in machine learning pipelines and the emergence of automated feature generation techniques such as deep learning, has further exacerbated the problem. To address this problem, the authors present DARWIN, an interactive system designed to alleviate the task of writing rules for labeling text data in weakly-supervised settings that enables annotators to generate weakly-supervised labels efficiently and with a small cost.

2. PROBLEM DEFINITION

For a more formally definition of the problem, given a labeling task, our goal is to find a set R of adequately accurate labeling rules such that the union of their coverage denoted as $P = \cup_{r \in R} C_r$, would have a high recall (i.e., to contain majority of the positive instances in the corpus) and all rules $r \in R$ are accurate. We would like to maximize the recall of set P without posing too many queries to the oracle.

3. PRIOR WORK

A subset of existing frameworks, such as Snorkel, rely on domain experts to provide a set of labeling rules which can be a tedious and time-consuming task. In contrast, other frameworks aim to automatically mine useful rules using further supervision. For instance, Snuba circumvents dependence on domain experts by requiring a labeled subset of the data, and then utilizing it to automatically derive labeling rules. Babble labble is another example which asks expert to label a few examples and explain their choice. However, in practice, the positive instances often make up only a tiny fraction of the entire corpus. Hence, labeling a random sample would not be sufficient to obtain enough positive instances.

4. SOLUTION

In this section, we describe the architecture of DARWIN which is illustrated in Figure 3. The pipeline is initialized with a seed labelling function or a couple of positive sentences. DARWIN learns a rudimentary classifier using these positive sentences and it is refined with evolving training data.

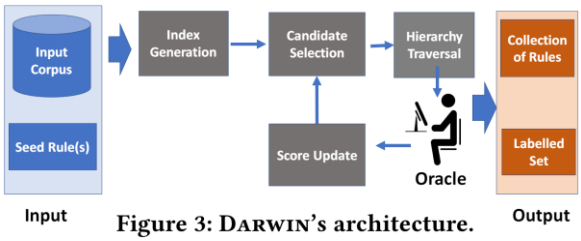

Figure 3: DARWIN's architecture.

In each iteration, DARWIN performs the following steps. First, the Candidate Generation component generates a small set of promising candidate rules (from the space of all possible rules), and organizes them in the form of a hierarchy H with the most generic rule at the top and the stricter ones at the bottom. Once the hierarchy is built, the Hierarchy Traversal component carefully navigates and evaluates the rules in the hierarchy to find the best candidate. This candidate is then presented to the annotator and scores are updated. Finally, the updated classifier and rule scores are sent back to hierarchy generation and traversal for the next iteration.

5. RESULTS

5.1 Comparison with existing work

As shown in Figure 6, this experiment validates that Snuba works well when the initial seed set has enough randomly chosen positives and does not generalize to rules that have limited evidence. On the other hand, DARWIN(HS) (HS represents the combination of LocalSearch and UniversalSearch traversal technique) identifies majority of the positives even when the pipeline is initialized with just 25 sentences and has good generalizability.

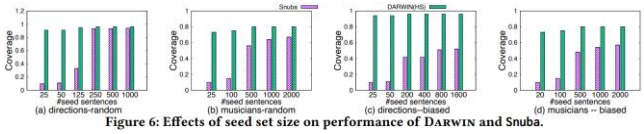

Figure 6: Effects of seed set size on performance of DARWIN and Snuba.

5.2 Quality of the classifier

As shown in Figure 7, the active learning technique suffers from poor F-score initially and improves gradually. Since AL generates very few training examples, the trained classifier is highly unstable and shows jittery F-score. The KS approach shows similar performance and performs comparable to AL. On the other hand, DARWIN based pipelines are much more stable in terms of F-score. The classifier that was trained with the labeled data generated by DARWIN pipelines always maintains a high precision.

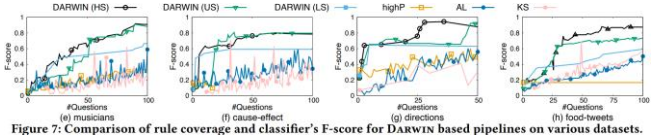

Figure 7: Comparison of rule coverage and classifier's F-score for DARWIN based pipelines on various datasets.

6. CRITIQUE

DARWIN achieve superior performance in labeling text data at weakly-supervised learning environments. One notable strength of DARWIN is that it can automatically mine useful rules with systems, while requiring far less labeled instances compared with other state-of-the-art frameworks, which rely heavily on pre-labeled data. Furthermore, this system requires experts to simply verify the suggested rules instead of manually writing rules or providing explanations for why a particular label is assigned.

7. EXTENSION

There have been several studies on using weakly-supervised labels in an optimal way, e.g. data fusion, truth discovery and

handling crowd error. To achieve better results of this system, DARWIN's generated rules can be further processed using these de-noising techniques.

Since DARWIN is effective in creating rules with minimal initial examples and iteratively refining them, this paper did not mention the ability for ongoing updates as new data evolves. The potential extension is to make the system continuously updating rules on its own, without the requirement to reinitialize the system with new seed labels.