Comparing Methods for Automatic Identification of Mislabeled Data

Jessica Sweeney

A thesis
submitted in partial fulfillment of the
requirements for the degree of

Master of Science

University of Washington

2021

Reading Committee:
Shane Steinert-Threlkeld, Chair
Swabha Swayamdipta

Program Authorized to Offer Degree:
Computational Linguistics

University of Washington

Abstract

Comparing Methods for Automatic Identification of Mislabeled Data

Jessica Sweeney

Chair of the Supervisory Committee:
Assistant Professor Shane Steinert-Threlkeld
Department of Linguistics

This thesis compares three methods for identifying mislabeled examples in datasets: Dataset Cartography (Swayamdipta et al. [2020]), Cleanlab, (Northcutt et al. [2021b]), and Ensembling (Brodley and Friedl [1999], Reiss et al. [2020]). Mislabeled examples in the training data of a dataset deteriorate the learning signal that models can use for the task, and mislabeled data in the test split prevent accurate assessment of a model's performance, so it is useful to have methods to identify and correct those labels. In order to compare the methods as directly as possible, we use the Multi-Genre Natural Language Inference corpus (MNLI) as the dataset that all methods will inspect for mislabeled examples (Williams et al. [2018]). We choose RoBERTa-large (Liu et al. [2019]) as the model which generates information about MNLI to be used as input for each method, and we compare the lists of mislabeled examples predicted by the three methods. Manual inspection of a subset of the data reveals that Dataset Cartography has the highest accuracy in identifying truly mislabeled examples, followed by Cleanlab, followed by Ensembling. The methods share about half of their total flagged examples in common, and all produce around the same number of examples (approximately 20k). They all flag examples labeled "neutral" at about twice the frequency of the other labels in MNLI. When data is removed from the original training set according to the lists produced by the models, without manual inspection, performance is reduced on a challenge test set (HANS), though Cartography and Ensembling's performances are reduced

slightly less than Cleanlab's. Overall, Dataset Cartography seems to be the highest accuracy method in this particular context, and would be most likely to reduce the amount of manual relabeling needed to be done during the process of cleaning a dataset's labels.

TABLE OF CONTENTS

LIST OF FIGURES

ACKNOWLEDGMENTS

I wish to thank the various people who have supported and encouraged me throughout the writing of this thesis and in the completion of the CLMS program: Shane Steinert-Threlkeld, for being an excellent advisor and professor; Swabha Swayamdipta, for her thoughtful comments and feedback; Devin Brown and Katya Simpson, for being great study partners and even better friends; the CLMBR group, for engaging discussions about research; and all of my friends and family who listened to me talk about this thesis for several months straight, providing support and guidance along the way.

DEDICATION

To my parents, Susanna and Peter Sweeney

Chapter 1

INTRODUCTION

Labeled data is an extremely valuable resource as part of a well-constructed dataset that focuses on a linguistic task, which helps drive the improved performance of language models (LMs) that are fine-tuned on that particular task. Natural language understanding datasets in particular present a challenge, both for LMs and dataset builders. NLU datasets test a model's ability to use semantic information from the input to perform a task. MNLI and natural language inference tasks in general are NLU tasks that ask models to decide whether one sentence contradicts, is neutral to, or entails another (Williams et al. [2018]). The General Language Understanding Evaluation (GLUE) benchmark contains several NLI tasks, including MNLI, as well as question-answering, sentiment analysis, and paraphrasing tasks, which all require a model to build a representation of the semantics of its input in order to produce the correct output (Wang et al. [2019]).

Because of the semantic nature of NLU tasks, NLU datasets require input from humans for construction: producing text, assigning labels, or both. This human input introduces a potential failure point for these datasets, since humans will occasionally make mistakes, either mechanically or in their own comprehension of the task. For example, BoolQ, a large question-answering dataset scraped from Google searches, reports a 90% accuracy of its human annotators (Clark et al. [2019]). This number represents a ceiling on how well we are able to understand the model's performance, since it affects the quality of the training data and the interpretability of the model's accuracy on the test set. While 100% ground truth accuracy is impossible for some tasks, due to ambiguity or disagreement among human annotators, it is still desirable to separate examples that are correctly labeled with high confidence from those that might be mislabeled or too ambiguous to label. This would allow

dataset creators to either remove or relabel those examples without manually inspecting the entire dataset, allowing it to be used to make more interpretable assessments of a model's performance.

The accuracy of the gold standard labels in the train split of a dataset affects whether the model has enough signal to learn a decision boundary for the task in question. Training sets are often large and looked at by fewer sets of eyes than test sets, due to the human labor cost of creating a dataset, but systematic sources of error in training sets can reduce model performance (Reiss et al. [2020]).

The accuracy of gold labels in the test set is important for correctly evaluating the performance of the model. It is often implicitly assumed that the accuracy of the labels in the test set is 100%, and therefore that 100% accuracy on the test set is the goal of model improvement efforts. This assumption becomes a problem when a dataset's ground truth accuracy is less-than-perfect itself. (A dataset's ground truth accuracy is technically inaccessible, but it can be approached with inter-annotator agreement among experts.) If only 90% of the labels are correct in terms of the task specification, it is not desirable for models to exceed that accuracy. That would be an indication that it is overfitting to the labelling errors, which in turn is an indication that it has not learned the true decision boundary that solves the task. It is also not necessary for datasets to be entirely free of errors or ambiguity, but it is important to be aware of the percentage of the dataset that may contain noise.

Despite the knowledge that datasets may contain errors that can affect the way we ought to interpret them, many are still published without comment on possible sources of error, and some datasets (such as the Amazon Product Reviews corpus) have "naturally occurring labels" that are not reflective of the task specification (e.g. a positive sentiment review with a 1-star rating) (Northcutt et al. [2021a]). Verifying a dataset's labels is a complex task involving human labor and thus is time-consuming and expensive. This motivates the development of methods that can automatically identify potentially mislabeled examples, speeding up the process of review by reducing the number of datapoints that need to be

evaluated by a human. If these automatic methods have high accuracy in identifying mislabeled data, flagged examples could even be removed without inspection, accepting the cost of removing small amounts of good data in order to remove the bad data.

Because of the potential value of methods for identifying mislabeled data, this thesis aims to compare the performance of three of these existing methods. The papers that introduce the methods looked at in this study provide data about their performance in various settings, but there are no experiments across the three methods that can be used to directly compare their efficacy, since the methods all present results on different datasets with different models. This study attempts to apply three methods for identifying mislabeled examples to the same dataset (MNLI), using the same model (RoBERTa-Large) and hyperparameters for training, in order to rank them by their performance. Their differences in performance in this setting can be applied to other NLI datasets, and potentially other model architectures. This information would allow more informed decisions to be made when selecting a method for cleaning data in practice.

1.1 Research Questions

Given a set of flagged examples from each method, there are several questions that can be asked that help compare their efficacy:

1.1.1 Accuracy

How many of the examples flagged by each method are "truly" wrong, that is, what is the accuracy of each method? Since we are comparing the given labels to a latent ground truth, which would disagree with the given labels for genuinely mislabeled examples, we need to relabel the instances we are interested in, and perform an accuracy analysis on this subset. A single annotator will relabel the examples without knowledge of their gold labels.

1.1.2 Agreement Between Methods

For the methods that rank examples based on how likely they are to be mislabeled, how similar are their rankings? Since all three methods purport to be identifying members of the same mislabeled set, and are assigning them a probability as to how likely they are to be a true member of that set, we might expect them to come up with similar rankings if they are identifying a true latent feature of the data.

1.1.3 Effect on Model Performance

If the flagged examples were removed without any supervision, how would a model trained on this cleaned data perform on its original test set? On a challenge set? The cleaned data will contain less noise than its original counterpart, which we would expect to provide a clearer training signal to the model, which in turn would boost performance on the task. However, the methods might flag correctly labeled data that would also provide a valuable training signal to the model, and testing it on a challenge set will allow us to evaluate whether performance changes on a subset of the data with particular characteristics.

Since the use case of these methods is to reduce human effort in cleaning datasets, one possible application is to remove all flagged data without manually relabeling it, to ensure that the remaining data is as clean as possible. This changes the distribution of the data, and the imperfections of each method (the properly labeled examples that are included in their flagged sets) might affect the training signal that remains for learning when the "mislabeled" examples are removed. It is important to know if this negatively effects model performance, especially on more challenging examples.

Chapter 2 provides an overview of relevant literature, including discussion of the three methods that will be focused on in this thesis, as well as methods outside the scope of this research and methods for evaluation. In the section after, I describe the methods I use and the experimental conditions I set, as well as discussion of how I will perform evaluation. Next, I present results from the experiments, followed by an analysis and discussion of those

results.

Chapter 2

BACKGROUND AND LITERATURE REVIEW

The question of how to clean mislabeled or noisy data has been under discussion for a long time, but in the past few years there has been growing interest in the quality and character of popular datasets that are used to evaluate and compare the performance of large language models.

In response to this, several methods have been presented to clean datasets by automatically identifying potentially mislabeled instances. The current chapter provides a survey of these methods. Ensembling, Dataset Cartography, and Cleanlab are the three methods for flagging mislabeled data that will be compared to each other in this thesis. The other methods in this literature review (in Section 2.5) will not be implemented here, but are included for context.

This literature review also contains a closer look at HANS, the challenge set for MNLI, that will be used as a secondary evaluation of the data cleaning methods in this thesis.

2.1 Ensembling

An early look at methods of identifying mislabeled data can be found in Brodley and Friedl [1999]. They propose filtering training data by using an ensemble of classifiers and n-fold cross-validation. For each of the n different subsets of the training data, m classifiers are trained on the other $n - 1$ parts, and then are used to label the held-out data, which produces m labels for each example in the dataset. There are multiple strategies for using this information; for instance, Brodley and Friedl [1999] distinguish between the consensus filter, which only flags an example as mislabeled if all m models to predict a label other than the gold label, and the majority vote filter, which requires the majority (but not necessarily

all) of the ensemble to disagree with the gold label.

They measured the efficacy of the filters by randomly permuting the training data labels and calculating precision and recall of the filters on the corrupted data. In general, the consensus filter was conservative in flagging an example as mislabeled, so it had high precision but low recall. In comparison, the majority vote filter found a higher proportion of the mislabeled examples, but tended to discard many more correctly labeled examples. They recognize that their algorithm cannot distinguish between truly mislabeled examples and examples that are simply exceptions to a general rule, or difficult for tfhe models to learn.

The ensemble filtering method was used in a recent paper by Reiss et al. [2020] to identify mislabeled entities in the CoNLL-2003 dataset (Tjong Kim Sang and De Meulder [2003]). They used a BERT model (Devlin et al. [2019]) fine-tuned on the corpus to obtain word embeddings for the full dataset, and then trained an ensemble of linear models on cross-validation folds of the training set embeddings to identify examples that were mislabeled by a majority of models. They used human annotators to validate and correct the examples that were flagged by the ensemble, 34% of which were incorrectly labelled. They find that models trained on the corrected training data perform better on the corrected test data than the same models that were trained on the original training data.

2.2 Dataset Cartography

Swayamdipta et al. [2020] present Data Maps, a tool that tracks the training dynamics of a model being fine-tuned on a dataset in order to produce a visualization of that dataset. "Training dynamics" refers to two values that are tracked for each example over the course of the epochs needed for training: the average confidence assigned to an example's gold label, and the variability of that confidence across all epochs. Figure 2.1 shows a Data Map with confidence on the vertical axis and variability on the horizontal axis; these maps reveal semi-distinct regions of the dataset as a function of these two numbers. The lower-left region, the "hard-to-learn" examples, is characterized by consistently low-confidence examples, and the authors argue that many of the mislabeled examples in a dataset are to be found in this

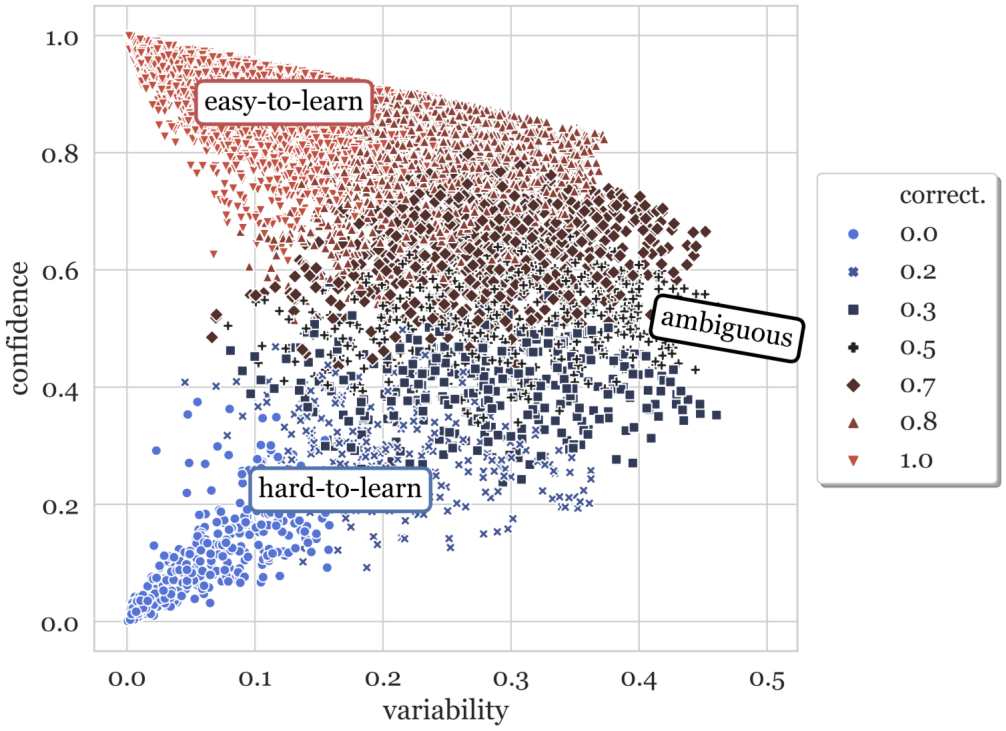


Figure 2.1: A Data Map for the SNLI train set, based on a RoBERTa-large classifier.

region of the graph.

They propose a method for extracting likely mislabeled examples from the dataset using these two measures. They add artificial noise to the dataset in question by flipping the gold labels and retrain the model on the noisy data. They then train a linear classifier to predict whether an example from this dataset is mislabeled (i.e. label-flipped) or not, using the confidence score as the single feature for each example. This trained classifier can then be used on the confidence scores from the original dataset (which were computed without added noise), to identify which of those examples might be mislabeled. They find that despite differences in label balance between the classifier's training data and the dataset itself, the classifier does not over-predict "mislabeled", which are much less frequent in the real data

than in the training data for the classifier.

To test the success of the classifier on data that has not been artificially noised, they randomly selected 50 instances of each predicted class (correctly labeled and mislabeled), and used two human evaluators to re-annotate those examples, using discussion to come to agree on 96% of the instances they evaluated. Using these annotations as a new gold standard for those dataset examples, they find that in WinoGrande (Sakaguchi et al. [2019]), 67% of the instances predicted as noisy by the classifier are either mislabeled or ambiguous, and 76% of those instances in SNLI are indeed mislabeled.

The authors also try to use variability, the other measure tracked by the Data Maps, as the feature used by the linear classifier, but find much lower performance on the artificial data. Confidence (i.e. the probability assigned to the gold label by the model) is used by all three methods looked at in this thesis, but Cartography is the only method that uses average confidence across the training epochs of a model, as opposed to the probability assigned after training and during inference.

2.3 Confident Learning

Northcutt et al. [2021b] present a novel approach to identifying mislabeled instances in datasets called confident learning (CL), which estimates the joint distribution between the noisy (given) labels of a dataset and the uncorrupted (unknown) labels. This method constructs a matrix C that is $m \times m$, where m is the number of labels in the dataset. Each row represents the gold label of an example, and each column represents the latent, "true" label. The count of a cell in this matrix is incremented for each dataset example whose predicted label probability is greater than or equal to the average predicted probability for its gold label across all dataset examples. For example, if the average confidence assigned to the "entailment" label for all of the examples with "entailment" as their gold label in MNLI is 85%, then an example for which the model predicts "entailment" with 90% probability whose gold label is indeed "entailment" would be added to the count of the $C_{entail.,entail.}$ cell. However, if the gold label of an example is "entailment" but the model predicts "contradiction" with

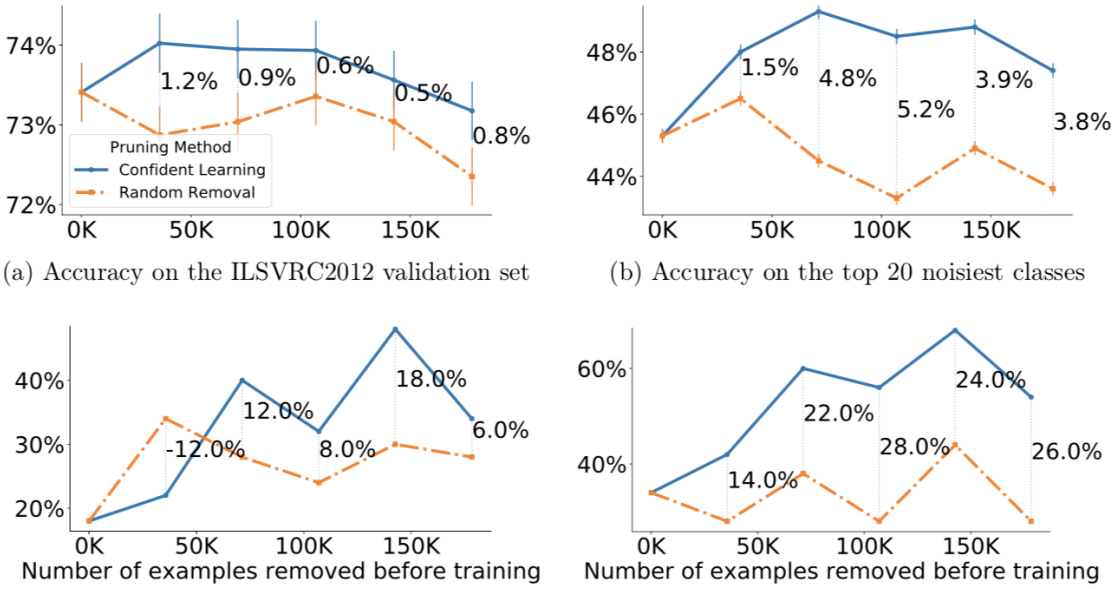
90% confidence, that would be added to the count of the $C_{entail.,contra.}$ cell, and would represent a potentially mislabeled example. This matrix is then normalized so that the row-sums match the observed marginals and the entire distribution sums to 1. This distribution is used to estimate the prevalence of noisy data in the dataset, prune those examples, and rank the remaining instances in order of model confidence in the label.

In Northcutt et al. [2021a], the authors use a logistic regression classifier with fastText tri-gram embeddings to identify mislabeled reviews in the Amazon Reviews and the IMDb Movie Review datasets (Northcutt et al. [2021a]). From the IMDb dataset, which is smaller, they identified 1,310 potentially mislabeled instances using CL, 725 (55.3%) of which were validated by MTurk workers as mislabeled. The Amazon dataset was much larger, and identified 500k errors. They chose 1,000 instances to be checked by MTurk workers, 732 (73.2%) of which were validated as mislabeled. Across 10 datasets, including Amazon and IMDb as well as image and audio datasets, the precision of their method varied from 15% to 89%, depending on variables like the size, difficulty, and quality of labelling of the original dataset.

2.4 Assessing Effectiveness

Many of the datasets looked at in these papers produce potentially mislabeled sets that are too large to inspect by hand. For example, it's possible to review and correct some percentage of the 500k errors flagged in the Amazon Reviews dataset, but the process will be costly and time-consuming and can be another source of errors (especially in a multi-class task). One alternative is to remove the flagged data, rather than cleaning it. This method is tried by Northcutt et al. [2021b], who report validation set accuracy for models trained on datasets with varying percentages of flagged data removed, shown in Figure 2.2.

They find that removing the flagged data increases model performance on the validation set in general and on noisy examples in particular. This is consistent with another early paper in the area of denoising training data, Jiang and Zhou [2004b], which looks at the Depuration algorithm and suggests improvements. The Depuration algorithm (introduced in Barandela



(a) Accuracy on the ILSVRC2012 validation set

(b) Accuracy on the top 20 noisiest classes

(c) Accuracy on the noisiest class: foxhound

(d) Accuracy on known erroneous class: maillot

Figure 2.2: Accuracy measures as reported by Northcutt et al. when increasing percentages of the flagged data are removed from the training set.

and Gasca [2000]) uses a variation of the k Nearest Neighbors (kNN) algorithm to either remove or relabel potentially mislabeled instances. If the majority of the k nearest neighbors of an instance are labeled differently from that instance, it is relabeled to the majority class. If there is no majority of the k nearest neighbors, the instance is removed. Jiang and Zhou [2004b] find that separating out these two different aspects of the Depuration algorithm (removing and relabelling) can provide higher accuracy on the test set than doing both. Specifically, the algorithm that removes the data outperforms Depuration (both removing and relabelling) and the algorithm that only relabels in most cases.

2.5 Other Methods

Esuli and Sebastiani [2009] present three different techniques for performing Training Data Cleaning (TDC). They define a good TDC technique as one that top-ranks the training examples that are most likely to be misclassified, to reduce annotator effort during the relabeling process. They focus on TDC techniques for text classification, which is a classification setting consisting of a predefined set of categories for which each document has either a positive or negative label.

The first technique they call the confidence-based technique, or CON, for short. CON consists of training a classifier and producing a label and a confidence value for each example in the dataset, then top-ranking the misclassified examples, in decreasing order of their assigned confidence, and then appending the correctly classified examples in increasing order of assigned confidence. This requires the cross-validation that is used in Ensembling, but only one classifier predicts a label for each example. The second technique is the nearest-neighbors technique (NN), which ranks training examples by how consistent their label is with their surrounding dataset examples. This involves computing a distance-weighted measure of the difference between the labels of the neighboring data points and the example in question. The third technique is the committee-based technique (COM), which is similar to the Ensembling technique in this paper, in which the committee of classifiers casts a weighted vote for a label for an example, weighted by their confidence in the label. They find that the CON and NN

techniques perform the best across different datasets and class frequency.

Extending the work of Esuli and Sebastiani [2009], Malik and Bhardwaj [2011] present Automatic Training Data Cleaning (ATDC), a method for cleaning training data that relies on some subset of the training data having high-quality labels. The input to this method is a large automatically- or noisily-labeled training set, with another training set with high quality labels. The output is a subset of the large noisy set that can be used to augment the high quality labels; this subset has had its labels validated by the ATDC method.

Their algorithm first splits the high-quality training data into two sets, train and test. The train set is used to find examples from the low-quality dataset that are most likely to be correctly labeled, and the test set is used to evaluate those examples' effect on classification accuracy when they are used to augment the train set. The train set is split into k sets, and the algorithm iterates k times, with 1 of the sets held out for evaluation at each iteration. With the other $k-1$ sets, they train a binary classifier and obtain its baseline accuracy on the held out set. This classifier is then used to generate labels for the the low-quality dataset, which is one form of evidence they use to validate the given label. The other form of evidence they use is a label generated by clustering the high-quality data with the noisy data, and assigning labels to the noisy data based on its nearest-neighbors from the high-quality set. These two pieces of evidence, along with the original (noisy) class-labels can be used in an evidence evaluation scheme to decide whether to include a low-quality example in the augmented set. This scheme can be more or less strict; Table 2.1 shows the evidence evaluation schemes used in the original paper, in descending order of strictness.

They find that taking a subset of the examples that were selected by the strictest evaluation scheme to augment the high-quality data yields the best classification performance, and that including data from the less strict schemes produces a less reliable classifier. MNLI does not have a subset with high quality labels, which makes this method outside of the scope of this research.

Scheme	$C_x \in$ Orig. Labels	Positive Evidence
S1	Yes	Classifier and Clustering
S2	Yes	Classifier or Clustering
S3	No	Classifier and Clustering

Table 2.1: Schemes for classifying an example as high-quality. C_x varies across all potential labels an example can have. If C_x is its original label, the strictest scheme admits it as a high-quality example of that label only if the clustering and classifying algorithms also assign it that label. The second-strictest scheme accepts either of those as positive evidence if C_x is the gold label. If C_x is not the gold label, then both the clustering and classifying algorithms must assign it as a label to the example for it to be included as a positive example of that label.

2.6 MNLI

The Multi-Genre Natural Language Inference corpus (MNLI) was constructed by Williams et al. [2018], the same group that produced the Stanford NLI corpus (SNLI) (Bowman et al. [2015]), and it was designed to improve upon the coverage and difficulty of SNLI. MNLI contains data from ten distinct genres of both written and spoken English, and its 433k pairs of sentences have all been human-annotated with one of three labels: entailment, contradiction, or neutral.

The task is to assign each pair of sentences with a label based on their relationship to each other's truth values; that is, whether they entail one another. If the first sentence in a pair (the "premise") entails the second sentence (the "hypothesis"), then the hypothesis is always true if the premise is true. It is a contradiction if the hypothesis cannot be true when the premise is true, and they are neutral if their truth values do not depend on each other.

This task is important for a model to be able to perform because it requires determining the truth values of sentences and how they relate to each other, which requires building

general-purpose and versatile semantic representations of the sentences. These representations can be transferred to or used for other tasks, such as zero-shot classification.

MNLI is a large and well-known NLI dataset, so its accuracy is important to the field. An analysis by Gururangan et al. [2018] shows that the dataset contains artifacts that make the task easier for models that pick up on heuristics like word overlap or negation. If, in addition to these heuristic issues, the dataset contains a high percentage of mislabeled or ambiguous data, that would be important for further informing our interpretation of results obtained on MNLI.

The human annotators were crowdworkers who were given instructions for producing hypothesis sentences based on the premise sentences and were also provided with an FAQ and example sentences to help them understand the task. They were prompted with a premise and a label and were asked to write a sentence that satisfied that entailment relationship. For the train set, this label was taken to be the gold label for the sentence pair. For the validation and test sets, the pairs of sentences were given to validators who were asked to provide a single label for the pair. Four workers labelled each example, with the fifth label being the original hypothesis writer's. A minimum of a three-vote consensus was required for each example to be included in the corpus with a gold label.

The MNLI paper provides an 'Agreement' measure for each genre of the corpus, which represents the percentage of individual labels that match the gold label across validated examples. For the corpus overall, this Agreement measure is 88%, meaning that the average agreement on any given example is between four and all five annotators agreeing on the gold label.

2.7 HANS

The Heuristic Analysis of Natural Language Inference Systems (HANS) dataset, created by McCoy et al. [2019], will be used as an additional evaluation of these methods when used in an entirely automatic context, that is, without human relabeling. One of the benefits of using automatic methods of identifying mislabeled examples is lowering the human cost of

Heuristic	**Premise**	**Hypothesis**
Lexical Overlap	The doctors visited the lawyer.	The lawyer visited the doctors.
Subsequence	The judges heard the actors resigned.	The judges heard the actors.
Constituent	If the actor slept, the judge saw the artist.	The actor slept.

Table 2.2: Examples of sentences that represent the three heuristics tested by the HANS dataset. All three sentences have a non-entailment relationship, but we would expect a model relying on the heuristics to incorrectly label all examples as *entailment*.

cleaning datasets, so the ideal use case of these methods is discarding the flagged data sight unseen. However, if this 'cleaned' data is used to train a model to perform well on the NLI task, it might be missing crucial examples that were flagged as wrong by these unsupervised methods but are simply difficult or counterexamples/exceptions to a general rule. HANS is a challenge dataset for models trained on NLI datasets, and shows that many models rely on heuristics for making decisions.

HANS is a constructed dataset, relying on templates to produce its 30k examples. These templates are constructed so that the hypothesis and premise will overlap in either words, subsequences, or constituents. These overlap heuristics, in the original MNLI dataset, are far more likely to be labeled as "entailment" than "neutral" or "contradiction". Table 2.2 shows examples of sentences from HANS that follow the heuristics but are not labeled "entailment". McCoy et al. [2019] find that a BERT model (Devlin et al. [2019]) trained on the MNLI train set and evaluated using HANS will get more than 75% of the overlap-non-entailed examples wrong, showing that models trained on the full train set learn heuristics that do not hold in all cases.

The current research will look at Ensembling, Dataset Cartography, and Cleanlab, comparing them using the methods described in the next section, using HANS as an evaluation metric in addition to the MNLI test set. The other methods described in this literature

review represent future directions for further comparison between methods.

Chapter 3

METHODS

The current study implements the methods described in Reiss et al. [2020], Swayamdipta et al. [2020], and Northcutt et al. [2021b], applied to the MNLI dataset, with the goal of comparing their performance in similar environments. Each paper reports their method's performance on one or more datasets, but there is no common dataset between them that can be used as the basis of direct comparison; this study attempts to produce a direct comparison that can serve as a useful foundation for deciding between these methods in a practical setting.

3.1 Overview and Experimental Conditions

The methodology laid out here is designed to evaluate these three methods while keeping the conditions of their implementations as similar as possible. Because of the characteristics of these methods, there are certain aspects that must differ, but we attempt to keep those differences minimal and will indicate them in this section when describing their implementation.

3.2 Dataset

The Multi-Genre Natural Language Inference corpus (MNLI) was constructed by Williams et al. [2018], the same group that produced the Stanford NLI corpus (SNLI) (Bowman et al. [2015]), and it was designed to improve upon the coverage and difficulty of SNLI. MNLI contains data from ten distinct genres of both written and spoken English, and its 433k pairs of sentences have all been human-annotated with one of three labels: entailment, contradiction, or neutral.

In this work I will be using the entirety of the MNLI v1.0 dataset. Like other work on the MNLI dataset, the implementation for the Dataset Cartography method does a small amount of preprocessing, which discards examples that do not have a gold label (because the annotators did not reach consensus) and also discards examples whose ID numbers clash with another example in the dataset. This process still yields approximately 400k examples, and represents a clean set of examples with gold labels that can be easily distinguished from each other, so this cleaned data is used as input to the other two methods.

3.2.1 Presence of Noise

Since knowing whether an example is mislabeled requires manual inspection and is sometimes impossible because of ambiguity, one way to assess the performance of a method that claims to identify mislabeled examples is to randomly flip a subset of the labels in the dataset. This artificially produces mislabeled examples, assuming the original labels were correct. Swayamdipta et al. [2020] use this method to assess the effectiveness of their method.

For our study, we prefer to avoid introducing artificial noise in this way because it creates mislabeled examples randomly, rather than as a function of the content of the example itself. In the train set of MNLI, the labels are assigned based on the prompt the crowdworker was given for a premise. Whether or not the crowdworker was able to write a hypothesis that accurately entails, contradicts, or is neutral to a premise depends on the premise itself (e.g. how long it is, how ambiguous it is, etc). In the test set, since MNLI's labels are based on an aggregation of 5 different human annotations, labeling errors are also unlikely to be the result of random noise (which would require 3 of the annotators to accidentally select a label they did not intend). Rather, in order for a majority of the annotators to converge on an incorrect label, the example would likely have to be difficult or ambiguous enough to make the incorrect label seem plausible. Since naturally mislabeled examples are more difficult (as evidenced by multiple annotators mislabeling them), models will have lower confidence in their predictions. Since the methods we are comparing all use confidence and variability of prediction in their selection of mislabeled examples, we prefer the distribution

of those examples in our experiments to match the naturally occurring distribution of errors in datasets. That is, a model's confidence in a straightforward and artificially label-flipped example will be higher than in an ambiguous example that was mislabeled by the annotators themselves.

There is evidence that many NLI examples are inherently ambiguous. Pavlick and Kwiatkowski [2019] show that when annotators are able to choose a value on a scale from entailment to contradiction, their judgments can form multiple distributions, indicating reasonable disagreement can exist between annotators on some NLI examples. We would want our methods to pick up on these ambiguous examples, in case we want to remove them from the dataset to ensure a clearer task specification, or so that they can be relabeled with both possible labels. Since sentence pairs like this would not resemble the artificially label-flipped examples (which models will have higher confidence on), we don't want to use performance on artificial examples to draw conclusions about these methods. For these reasons we focus on keeping conditions as close as possible to real application, and so we do not introduce artificial noise to the dataset.

3.3 Tools and Methods

This thesis compares 3 different methods. The details of their implementation in this study are described in this section.

3.3.1 Dataset Cartography

Dataset Cartography is a tool written by Swayamdipta et al. [2020], designed for the purpose of analyzing datasets. The main function of the tool is to produce maps of data points, with one axis representing the variability of an example (how consistently it was given the same label across epochs) and confidence (the average confidence of the model's prediction across epochs). This produces a graph of the dataset with three regions: easy-to-learn, hard-to-learn, and ambiguous.

In their paper, they make the claim that the hard-to-learn region of this map contains

Hyperparameter	Setting
Epochs	12
Optimizer	AdamW
Learning rate	1.099e-5
Batch size	96

Table 3.1: Hyperparameters used to train RoBERTa-Large model on MNLI.

most of the mislabeled examples. These datapoints have low confidence and low variability, meaning the model very consistently predicted the incorrect label, and had low average confidence in that label across epochs. The paper uses a classifier trained on artificially noised data in order to select examples from the unnoised set that are potentially mislabeled. The paper finds that using this method on SNLI, a smaller dataset than MNLI with a more restricted domain, 76% of the examples flagged as noisy by this method are actually mislabeled or ambiguous, as agreed upon by two annotators.

We use a slightly different filtering method to produce our set of flagged examples ranked by likelihood of being mislabeled. Using their claim that the hard-to-learn region contains most of the mislabeled examples, once the model was trained, we used their code for filtering the data to produce an ordered list of examples using the information from the model's training dynamics. We select all examples with less than 0.05 average confidence across all epochs, and these constitute our set of flagged examples for analysis. We used the configuration settings from their paper to train the RoBERTa model on the MNLI train set, which are shown in Table 3.1. All other hyperparameters are unchanged from the library we used to implement the method. [1]

[1]The Dataset Cartography code can be found here.

3.3.2 Cleanlab

This method requires predictions to be made on the data in an out-of-sample way, meaning a model had to be trained on a subset of the data and then used to make predictions on the held-out data. The trained model can be of any kind (a simple linear classifier would work), but to increase the similarity between the three methods, I fine-tuned four RoBERTa models on 3/4 of the MNLI data, and then used them to obtain predictions on the held-out 1/4 of the data. The RoBERTa models were fine-tuned using the code from the Dataset Cartography repository, which ensured that the hyperparameters were kept the same, and all the details of implementations were the same between the two methods.

Once the predictions were obtained from the four models, we used code provided by Northcutt et al. [2021b] to produce an ordered list of examples from the dataset, from most to least likely to be mislabeled. The inputs to their model were the final predictions made on each of the datapoints from the dataset, as well as the softmax probabilities the model assigned to them for each example. By using these probabilities to estimate the joint distribution of noisy observed labels and unknown true labels, Cleanlab estimates the number of noisy labels in the dataset and then identifies examples that are likely to belong to another class than their label, staying calibrated with the estimated level of noise.

3.3.3 Ensembling

This methods requires word embeddings produced by a model fine-tuned on the dataset in question. Since the Dataset Cartography method produces a fully fine-tuned model, we use the model that results from that method to produce word embeddings for the entire MNLI dataset. We concatenate the premise and hypothesis for each example in the dataset and tokenize them using the HuggingFace Transformers tokenizer (Wolf et al. [2020]), and pass each input through the fine-tuned model to obtain 1024-dimensional contextual embeddings for each word in the premise and hypothesis. We use average pooling over all tokens to get a single 1024-dimensional embedding for each example, which can be used as the input

features to a linear classifier.

In order to produce an ensemble of models from this single set of embeddings, we follow Reiss et al. [2020] and apply multiple different Gaussian random projections to the embeddings to reduce them to between 32 and 512 dimensions. By using two different projections for each of the dimensions, we create 10 sets of embeddings to train 10 different linear models. We use 10-fold cross-validation on the training set to get predictions from each model over the entirety of the training set.

In this method, flagged examples are ones on which the majority of the ensemble "voted" (via prediction) for a label other than the gold label. These examples are ordered into groups based on how many of the models agreed on the non-gold label. For example, if all 10 models agreed on "entailment" but the gold label was "contradiction", that represents a strong signal that the example is mislabeled. If for the same example, 9 models agreed on "entailment" but one predicted "neutral" or "contradiction", that example would be categorized in the group below. This creates 6 ordered groups for which at least half of the models disagree with the gold label, ranging from 5 models disagreeing to all 10 disagreeing. There is no further ordering for the examples within these groups.

3.4 Evaluation

To answer the first research question in Section 1.1.1 (What is the accuracy of each method?), a random subsample of the flagged examples from each method need to be relabeled by a human annotator. The examples will be shown without labels, the annotator will supply what they believe the gold label to be, and this will be checked against the given gold label in the dataset. If these match, the example will be considered correctly labeled, and if they do not, it will be considered mislabeled.

To answer the second research question in Section 1.1.2 (How similar are the methods' rankings?), the flagged examples will be compared between the two methods. This question is not necessarily concerned with whether the examples are truly mislabeled or not, but rather whether the two methods are picking up on the same examples. If they are, this

shows that they are picking out examples with similar characteristics, but if they aren't, it means they are sensitive to different aspects of the data, and using them together may be more useful than using one or the other.

To answer the third research question in Section 1.1.3 (How does removing flagged examples from the train set affect a model trained on the cleaned data?), we create three new training sets from the original data by removing the examples flagged by each of the methods, and re-fine-tune a RoBERTa model on each set of cleaned data. We then evaluate these models using HANS, a challenge set for MNLI, to see if performance drops from a model trained on the full train set. Since the methods flag different numbers of examples, we cannot attribute all of the performance change to a given method's accuracy —it may have just removed fewer total examples and therefore fewer correctly labeled examples. However, as a preliminary investigation, change in performance could motivate future work that controls further for amount of data removed. For example, if performance drops when data is removed, it is possible that the error detection methods have a tendency to remove data that would encourage out-of-distribution generalization and contradict heuristics that generally hold throughout the rest of the dataset.

3.4.1 HANS

Since the methods we are comparing here all rely on the model failing on examples in order to mark them as mislabeled, this increases the likelihood that the discarded data will include difficult, but correctly labeled, instances. Poor performance on HANS shows that trained models predict the incorrect label for premise-hypothesis pairs that contain word or subsequence overlap but do not entail each other. This implies that during training, models were performing poorly on examples in the train set that contain these heuristics but do not entail each other. Because of this, those examples will likely be flagged and removed by the cleaning methods in this study. Looking at the model's performance on HANS after the flagged data is removed will give some idea of the effect of removing data on how strongly the model learns heuristics (though these conclusions will be tentative, since the methods

remove different amounts of data, which can also have an effect on performance).

Chapter 4

RESULTS

4.1 Flagged Examples

The three methods produced a similar number of flagged examples, in the same order of magnitude and with a spread of 5k examples (Table 4.1 shows the total number of examples from each method). Table 4.1 also shows the label distribution of the flagged examples from each method. All of the methods flagged more examples with Neutral gold labels than Entailment or Contradiction, despite the full dataset being relatively label-balanced (∼130k of each label).

4.2 Precision

I sampled 102 examples (34 each of the 3 possible gold labels from MNLI) from each method and manually labeled them without knowledge of their given gold labels. Since Cleanlab and Dataset Cartography present their mislabeled examples in an ordered list, I selected these 102 examples randomly out of the first 500 returned by each method. Since there is no ordering in the Ensembling method outside of the number of models that agree on a non-gold

	Neutral	Entailment	Contradiction	Total
Cartography	10407	5613	3616	19636
Cleanlab	12346	5994	6347	24687
Ensembling	10444	5995	4245	20684

Table 4.1: Distribution of labels in the examples flagged by each method and counts of total examples flagged.

Gold label\Manual label	Contradiction	Entailment	Neutral
Contradiction	16	2	16
Entailment	8	18	8
Neutral	18	4	12

Table 4.2: Confusion matrix for the subset of Cleanlab examples that were relabeled by hand. The rows are the counts of examples by their gold labels from MNLI (and all sum to 34); the columns are the counts of examples by their manually assigned label. The count in the box in the Contradiction row and Entailment column is the number of examples whose gold label was Contradiction in the dataset, but was relabeled Entailment when manually inspected.

label, I randomly selected 102 examples from the set in which all of the models agree (which contained approximately 14k examples).

After hand-labeling by reading the hypothesis and premise and judging its entailment, I constructed confusion matrices for the mismatch in labels for each method. Because we are looking to see how many mislabeled examples were found by each method, we want the gold label from MNLI and the given label to disagree, and therefore it is better for the diagonal to have lower counts.

Table 4.3 shows that Dataset Cartography has the smallest diagonal of the three (sum = 12), followed by Cleanlab (sum = 46, Table 4.2), and then Ensembling (sum = 92, Table 4.4). Table 4.5 shows the overall accuracy on the 102 examples randomly selected from each method's flagged set; Dataset Cartography performed the best, with 88% of its examples being mislabeled. Cleanlab performed the second-highest, with just over half of its examples (55%) representing mislabeled datapoints, and Ensembling was low-performing, with only 9.8% of its flagged examples being mislabeled.

Gold label\Manual label	Contradiction	Entailment	Neutral
Contradiction	4	18	12
Entailment	15	7	12
Neutral	12	21	1

Table 4.3: Confusion matrix for the subset of Cartography examples that were relabeled by hand. The rows are the counts of examples by their gold labels (and all sum to 34); the columns are the counts of examples by their manually assigned label. The count in the box in the Contradiction row and Entailment column is the number of examples whose gold label was Contradiction in the dataset, but was relabeled Entailment when manually inspected.

Gold label\Manual label	Contradiction	Entailment	Neutral
Contradiction	33	0	1
Entailment	0	33	1
Neutral	1	7	26

Table 4.4: Confusion matrix for the subset of Ensembling examples that were relabeled by hand. The rows are the counts of examples by their gold labels (and all sum to 34); the columns are the counts of examples by their manually assigned label. The count in the box in the Contradiction row and Entailment column is the number of examples whose gold label was Contradiction in the dataset, but was relabeled Entailment when manually inspected.

Method	Correctly Flagged
Cartography	88%
Ensembling	9.8%
Cleanlab	55%

Table 4.5: Direct comparison between methods of what percentage of flagged examples were truly mislabeled from the manual labeling of the 102 examples selected for each method.

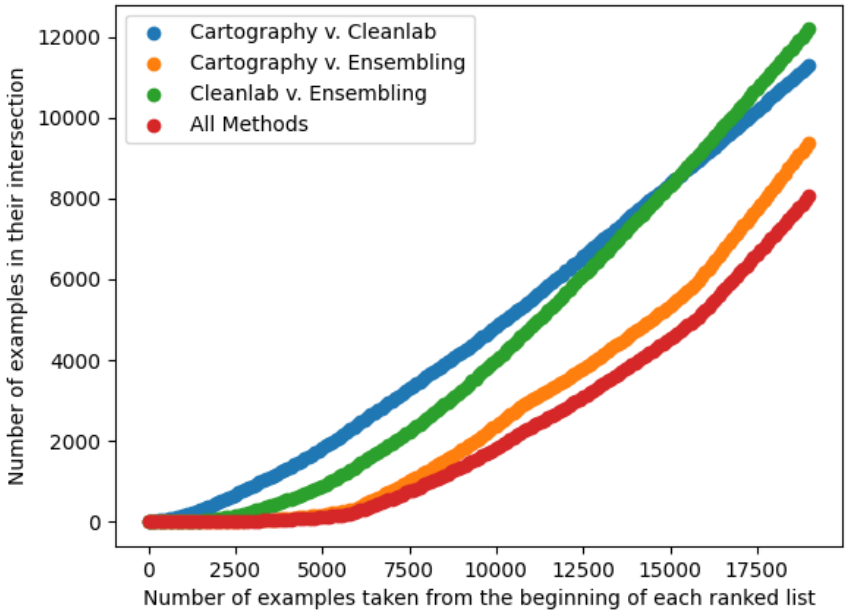


Figure 4.1: The number of examples shared between equal parts of the two or three ranked lists for each combination of methods in the legend. For example, at x=500, the y-value is the number of examples shared between the top 500 examples in each method's list.

Methods	**Total Shared Examples**
Cleanlab and Cartography	11223
Cleanlab and Ensembling	9271
Cartography and Ensembling	12086
All Methods	8052

Table 4.6: Number of examples that are common to the flagged sets of the pairs of methods in each row, as well as the examples that all three methods share in the final row of the table.

4.3 Agreement Between Methods

All of the methods produced roughly 20k examples, which might indicate that they are picking up on a similarly sized signal from the data. However, these 20k examples were not identical across methods, as shown in Table 4.6.

Because all of the methods yielded ordered lists (although Ensembling only provides large groupings), it is useful to check if they have similar orderings of their examples. To determine this, I iterated through progressively larger portions of each list, starting from the beginning, and compared the overlapping examples between all pairs of methods.

As Table 4.6 shows, Cartography and Ensembling shared the most examples overall (12,086), but Cleanlab and Cartography also share a comparable number of examples (11,223). Cleanlab and Ensembling disagree on more examples than either do with Cartography (only sharing 9,271).

The graph in Figure 4.1 helps explain the discrepancy between the methods' performance on the 100 randomly selected examples and their total shared examples. Since Cleanlab and Cartography share more than 50% of their flagged examples in common, it is unexpected that one would perform far better than the other on 100 examples. However, those 100 examples were chosen from the first 500 in each method, where the methods share fewer examples. You can see in Figure 4.1 that Cartography and Cleanlab share less than a fifth of the examples in their top 500.

Table 4.7 contains similarity measures between the two methods. Firts, we look at their rank correlation using Kendall's Tau. Ignoring examples that are not shared between the methods, we compare ordered lists of mislabeled examples produced by Cartography and Cleanlab, to see whether they generally agree on the probability that a given example is mislabeled. (Ensembling cannot be compared in this way because it does not rank the examples.)

The methods share 11,917 examples between them, and, preserving their ordering, we look at their rank correlation. Kendall's Tau for the two lists is 0.00248 (p=0.684), which

Kendall's Tau	Kendall's Tau (top 100)	RBO	RBO (top 500)
0.00248 (p=0.684)	0.0747 (p=0.271)	0.0269 ± 4.98e-12	0.0269 ± 5.138e-13

Table 4.7: Similarity measures between Cartography and Cleanlab, the two ranked methods.

is close to zero and therefore implies no correlation. Looking only at the top 100 shared examples, the tau is 0.0747 (p=0.271), which also implies no correlation.

Another measure, called Rank Biased Overlap (RBO), proposes to handle the problem of comparing ranked lists while taking into account depth of list (Kendall's Tau weights orderings at the beginning and end of the lists equivalently, but we want to weight the front of the list higher), and also taking into account that the lists may not contain the same set of elements (that is, their sets of elements overlap but are not identical) (Webber et al. [2010]). Comparing the Cartography and Cleanlab lists using the RBO measure, we see that they have a very low degree of similarity, with an RBO score of 0.0269 ± 4.98e-12. This takes into account the full list, which has a very long tail of lower-confidence predictions from both methods. Looking at only the top 500 examples from each method, we obtain an extremely similar RBO score of 0.0269 ± 5.138e-13. Both of these low scores imply very little similarity between the two lists.

4.4 Effect on Model Performance

Performance on HANS dropped for all three methods when when a RoBERTa-large model was trained solely on the data that was not flagged as mislabeled by the methods. Baseline accuracy on HANS for a RoBERTa-Large model trained on the entirety of MNLI is 79.95%, (60.8% on non-entailment examples, and 99.1% on entailment) and all of the methods perform below this. Table 4.4 shows the three methods' performances. Cartography and Ensembling perform comparably to each other but still favor "entailment" as a label for all examples, and Cleanlab follows the heuristics for all examples, never predicting "non-entailment".

	Cartography	Cleanlab	Ensembling
Mislabeled Data Removed	0.108 / 0.892	0.0 / 1.0	0.091 / 0.999
Random Data Removed	0.110 / 0.969	0.092 / 0.999	0.102 / 0.998

Table 4.8: Accuracy of RoBERTa-large models trained on data cleaned by the respective methods, tested on HANS. The first row is the accuracy of the models trained on the train set with the flagged data removed, the second row is the accuracy of the models trained on the train set with an equivalently sized random set of data removed. In each cell the accuracy on the non-entailment examples is on the left, and the accuracy on entailment examples is on the right.

In order to test whether the different amounts of data being removed by each method has an effect on model performance, we ran three baselines with random datapoints removed from the train set, of the same size as the flagged set removed by each method. The method that most clearly differs between removing mislabeled data and removing random data is Cleanlab, which has better performance on the non-entailment examples when random data is removed. This indicates that Cleanlab might be flagging examples that contradict the heuristics, removing some of the model's training signal that would allow it to perform well on HANS. Cartography and Ensembling have similar performance between mislabeled and randomly removed data, but performance is slightly higher when the data is removed randomly. More trials would need to be run to ascertain whether performance is consistently higher when random data is removed, but the current results imply that the two models' performances are comparable, if not strictly better.

We also ran these six models on the MNLI dev set, which was annotated by 5 crowdworkers. Once again, Cartography and Ensembling have comparable performance when a similarly sized amount of random data is removed from the training set, but Cleanlab's performance is far worse than an equally sized set of random data. This implies that Cleanlab is removing data that might be useful for generalization to the task.

	Cartography	Cleanlab	Ensembling
Mislabeled Data Removed	0.753	0.354	0.759
Random Data Removed	0.783	0.760	0.745

Table 4.9: Accuracy of RoBERTa-large models trained on data cleaned by the respective methods, tested on the MNLI dev set. The first row is the accuracy of the models trained on the train set with the flagged data removed, the second row is the accuracy of the models trained on the train set with an equivalently sized random set of data removed.

Chapter 5

DISCUSSION

The three methods produced a number of flagged examples within 5,051 examples of each other, though no two of the methods share more than 61% of their examples with each other. Since Cleanlab's method explicitly estimates the proportion of mislabeled data as part of its selection process, this similarity of size of flagged set in proportion to the entire dataset might reveal something about the prevalence of noise in MNLI. The other two methods produced a similar proportion to Cleanlab, and this could indicate that they are picking up on a signal shared by some subset of the data that is most likely to contain mislabeled examples, but which also includes some other examples (since none of the methods has 100% accuracy).

5.1 Accuracy

Cartography clearly outperformed the other two methods when randomly selecting from the top of its list of examples. Since the proportion of examples in its top 500 that it shares with the other methods is lower than for the list overall (see section 4.3), Cartography is likely to be the best method for reducing the manual label involved in relabeling. The higher rate of true positives in its list of possibly mislabeled examples means annotators will have to look at fewer correctly labeled examples when cleaning the data.

In this study, examples were relabeled by a single annotator, and therefore could also be incorrectly labeled, or represent only one of multiple possible interpretations (as discussed in Pavlick and Kwiatkowski [2019]). Collecting more annotations on subsets of the data would help ensure a better assessment of the accuracy of each method, and would help further identify which examples are ambiguous rather than mislabeled.

5.2 Agreement Between Methods

All of the methods have a higher rate of agreement in the tail end of their rankings than at the beginning, the highest-confidence predictions. This is especially prominent in comparing Cleanlab and Ensembling. Figure 4.1 shows that the two methods don't reach significant agreement until almost the 6,000th rank (though Ensembling's ranking is tiered, and its first 14k examples are equally ranked and therefore randomly ordered). In combination with the result that the accuracies of the two methods differ in their first 500 examples, this could indicate that the methods are picking up on different signals from the data.

The graph in Figure 4.1 also shows that Cartography and Cleanlab are choosing examples that are more similar to each other than either is to Ensembling, as evidenced by the shape of the graph being closer to linear at the beginning of the ranked list.

5.3 Effect on Model Performance

Removing the entire set of flagged examples seems to reduce performance on the challenge set HANS and is not recommended. Removing smaller portions of the data, from the beginning of the ranked list, may prove to be more effective in contexts in which manual relabeling is not possible. However, because removing any amount of data is guaranteed to remove some correctly labeled examples, future work would need to investigate whether the benefit of removing bad data can outweigh the harm of removing good data. In the absence of those results, manually labeling the data, beginning from the highest probability examples for the ranked methods, is the best course of action. The methods' predictions can also be combined by taking the intersection of their flagged examples, and that subset can be inspected. This allows for minimization of human labor, which can focus on the examples that are most likely to be truly mislabeled.

Chapter 6

CONCLUSION

This thesis attempts to compare as directly as possible three different methods for identifying mislabeled examples in datasets: Dataset Cartography, Cleanlab, and Ensembling. Gaining an understanding of which of these methods performs best, and in what contexts, is important for understanding when and where to deploy these methods in practice. We used MNLI as our dataset for testing these methods. MNLI and natural language inference in general is a suitable domain for the research questions in this thesis because it represents a general-purpose semantic task that humans do not attain 100% accuracy on. This ensured naturally-occurring errors in the dataset, and the semantic task presents more of a challenge to the model than a syntactic task (which models generally achieve higher accuracy on).

We fine-tuned a RoBERTa-Large model on the train set of MNLI and used the training dynamics of this model to produce Cartography's ranked list of examples. To obtain Cleanlab's mislabeled predictions, we used cross-validation to produce out-of-sample predictions for every example in the train set and used those predictions as input to the Cleanlab method, which produced a ranked list of examples. For Ensembling, we used the fine-tuned RoBERTa-Large model to produce embeddings over the entire dataset, and trained 10 linear classifiers on these embeddings to predict labels over the entire train set (using cross-validation again to obtain the labels in an out-of-sample way). The number of these classifiers that disagreed with the gold label produced a tiered ranking for the examples flagged by this method as incorrectly labeled. All methods produced approximately the same number of examples (20k), though they did not produce identical lists.

The performance of the three methods differed most drastically at the top of their ranked lists, where the examples that they proposed as most likely to be mislabeled were found.

Cartography performed the best according to the random subsample of examples from the top of its list. This fact should be taken into account when deciding which method to use to support human relabeling of examples, since Cartography is most likely to reduce the amount of time spent on correctly labeled examples.

The methods begin to share more examples in common as they move past their highest probability predictions. Interestingly, all methods yield a number of mislabeled examples within an order of magnitude of each other (a difference of 5,051 between the method with the most examples and one with the fewest), which suggests some uniformity in the magnitude of the signal that each method is picking up on. They also all flagged approximately twice as many "neutral"-labeled examples than either of "entailment" or "contradiction", which might imply a conservatism in the method, or perhaps a bias in the dataset towards "neutral" labels for ambiguous or difficult examples (which have lower confidence and are more likely to be flagged by the models). This is likely because the neutral label lies between entailment and contradiction semantically, and therefore has fewer extreme characteristics that mark it as an obvious member of its category, causing lower confidence on model predictions.

Future work might look at how these methods compare using different underlying models, or on different datasets. It might also include some of the methods from outside of the scope of the present work, to compare performance to a wider range of methods. Additionally, HANS is just one example of a challenge set, and it tests a very particular heuristic (namely, word overlap). Different datasets may have other heuristics that would lend themselves to being flagged by these methods, and an analysis of the kinds of examples on which performance drops when data is cleaned might be useful. A linguistic analysis of the flagged examples could also reveal patterns in the kinds of examples that tend to be difficult for models, which leads these methods to flag them as mislabeled.

The comparison between methods in this study might be of use in the process of deciding how best to clean a dataset that is suspected to contain mislabeled examples, especially if humans will be involved in the relabeling process (in which case one of the ranked methods will more efficiently reduce annotator labor). The direct application of this study's result is

limited to the MNLI dataset and the RoBERTa-Large model. Other BERT-based models might also elicit similar results, but models that are much smaller, larger, or of a different architecture than RoBERTa-Large have the potential to produce different results than in this study. However, our results can be tentatively extended to other NLI datasets using the RoBERTa model, since the similarity in task will keep the context similar to that of this study. Future work could explore these extensions and investigate where differences in method performance emerge.

BIBLIOGRAPHY

R. Barandela and E. Gasca. Decontamination of training samples for supervised pattern recognition methods. In *SSPR/SPR*, 2000.

S. R. Bowman, G. Angeli, C. Potts, and C. D. Manning. A large annotated corpus for learning natural language inference, 2015.

C. E. Brodley and M. A. Friedl. Identifying mislabeled training data. *Journal of Artificial Intelligence Research*, 11:131–167, Aug 1999. ISSN 1076-9757. doi: 10.1613/jair.606. URL http://dx.doi.org/10.1613/jair.606.

C. Clark, K. Lee, M.-W. Chang, T. Kwiatkowski, M. Collins, and K. Toutanova. Boolq: Exploring the surprising difficulty of natural yes/no questions, 2019.

J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019.

A. Esuli and F. Sebastiani. Training data cleaning for text classification. In L. Azzopardi, G. Kazai, S. Robertson, S. Rüger, M. Shokouhi, D. Song, and E. Yilmaz, editors, *Advances in Information Retrieval Theory*, pages 29–41, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg. ISBN 978-3-642-04417-5.

M. Gardner, Y. Artzi, V. Basmova, J. Berant, B. Bogin, S. Chen, P. Dasigi, D. Dua, Y. Elazar, A. Gottumukkala, et al. Evaluating nlp models via contrast sets. *arXiv preprint arXiv:2004.02709*, 2020.

S. Gururangan, S. Swayamdipta, O. Levy, R. Schwartz, S. R. Bowman, and N. A. Smith. Annotation artifacts in natural language inference data. *arXiv preprint arXiv:1803.02324*, 2018.

R. Jia and P. Liang. Adversarial examples for evaluating reading comprehension systems. *arXiv preprint arXiv:1707.07328*, 2017.

Y. Jiang and Z.-H. Zhou. Editing training data for knn classifiers with neural network ensemble. In F.-L. Yin, J. Wang, and C. Guo, editors, *Advances in Neural Networks – ISNN 2004*, pages 356–361, Berlin, Heidelberg, 2004a. Springer Berlin Heidelberg. ISBN 978-3-540-28647-9.

Y. Jiang and Z.-H. Zhou. Editing training data for knn classifiers with neural network ensemble. In *International symposium on neural networks*, pages 356–361. Springer, 2004b.

D. Kaushik and Z. C. Lipton. How much reading does reading comprehension require? a critical investigation of popular benchmarks. *arXiv preprint arXiv:1808.04926*, 2018.

Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov. Roberta: A robustly optimized bert pretraining approach, 2019.

H. H. Malik and V. S. Bhardwaj. Automatic training data cleaning for text classification. In *2011 IEEE 11th international conference on data mining workshops*, pages 442–449. IEEE, 2011.

R. T. McCoy, E. Pavlick, and T. Linzen. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. *arXiv preprint arXiv:1902.01007*, 2019.

C. G. Northcutt, A. Athalye, and J. Mueller. Pervasive label errors in test sets destabilize machine learning benchmarks, 2021a.

C. G. Northcutt, L. Jiang, and I. L. Chuang. Confident learning: Estimating uncertainty in dataset labels, 2021b.

E. Pavlick and T. Kwiatkowski. Inherent Disagreements in Human Textual Inferences. *Transactions of the Association for Computational Linguistics*, 7:677–694, 11 2019. ISSN 2307-387X. doi: 10.1162/tacl_a_00293. URL https://doi.org/10.1162/tacl_a_00293.

A. Ratner, S. H. Bach, H. Ehrenberg, J. Fries, S. Wu, and C. Ré. Snorkel: Rapid training data creation with weak supervision. *The VLDB Journal*, 29(2):709–730, 2020.

F. Reiss, H. Xu, B. Cutler, K. Muthuraman, and Z. Eichenberger. Identifying incorrect labels in the conll-2003 corpus. In *Proceedings of the 24th Conference on Computational Natural Language Learning*, pages 215–226, 2020.

K. Sakaguchi, R. L. Bras, C. Bhagavatula, and Y. Choi. Winogrande: An adversarial winograd schema challenge at scale, 2019.

S. Swayamdipta, R. Schwartz, N. Lourie, Y. Wang, H. Hajishirzi, N. A. Smith, and Y. Choi. Dataset cartography: Mapping and diagnosing datasets with training dynamics. *arXiv preprint arXiv:2009.10795*, 2020.

E. F. Tjong Kim Sang and F. De Meulder. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147, 2003. URL https://aclanthology.org/W03-0419.

A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. R. Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding, 2019.

W. Webber, A. Moffat, and J. Zobel. A similarity measure for indefinite rankings. *ACM Trans. Inf. Syst.*, 28(4), Nov. 2010. ISSN 1046-8188. doi: 10.1145/1852102.1852106. URL https://doi.org/10.1145/1852102.1852106.

A. Williams, N. Nangia, and S. Bowman. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122. Association for Computational Linguistics, 2018. URL http://aclweb.org/anthology/N18-1101.

T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. L. Scao, S. Gugger, M. Drame, Q. Lhoest, and A. M. Rush. Huggingface's transformers: State-of-the-art natural language processing, 2020.

X. Zhou, Y. Nie, H. Tan, and M. Bansal. The curse of performance instability in analysis datasets: Consequences, source, and suggestions, 2020.