Detecting Label Errors in Token Classification Data

Wei-Chen Wang[1,2], Jonas Mueller[1]
[1]Cleanlab, [2]MIT

Abstract: Mislabeled examples are a common issue in real-world data, particularly for tasks like token classification where many labels must be chosen on a fine-grained basis. Here we consider the task of finding sentences that contain label errors in token classification datasets. We study 11 different straightforward methods that score tokens/sentences based on the predicted class probabilities output by a (any) token classification model (trained via any procedure). In precision-recall evaluations based on real-world label errors in entity recognition data from CoNLL-2003, we identify a simple and effective method that consistently detects those sentences containing label errors when applied with different token classification models.

➤ Introduction

It has recently come to light that many supervised learning datasets contain numerous incorrectly labeled examples. To efficiently improve the quality of such data, **Label Error Detection (LED)** has emerged as a task of interest, in which algorithms flag examples whose labels are likely wrong for reviewers to inspect and correct. This paper considers LED for **token classification tasks** (such as entity recognition) in which each token in a sentence has been given its own class label. Here we propose **worst-token**, a method to score sentences based on the likelihood that they contain some mislabeled tokens, such that sentences can be effectively ranked for efficient label review. We evaluate the LED performance of this approach and others on real-world data with naturally occuring label errors, unlike many past LED evaluations based on synthetically-introduced label errors, for which conclusions may differ from real-world errors.

➤Methods

Given a sentence x, a token classification model $M(\cdot)$ outputs predicted probabilities $p = M(x)$ where p_{ij} is the probability that the ith token in sentence x belongs to class j. Throughout, we assume these **probabilities are out-of-sample.** Using p, we first consider evaluating the individual per-token labels. Here we apply effective LED methods for standard classification settings by treating each token as a separate independent instance. We compute a label quality score $q_i \in [0,1]$ for the ith token via one of the following options:

- self-confidence (sc): predicted probability of the given label for this token
$$q_i = p_{ik}$$
- normalized margin (nm):
$$q_i = p_{ik} - p_{i\tilde{k}} \text{ with } \tilde{k} = \text{argmax}_j \, \{p_{ij}\}$$
- confidence-weighted entropy (cwe):
$$q_i = \frac{p_{ik}}{H(p_i)} \text{ where } H(p_i) = -\frac{1}{\log K}\sum_{j=1}^{K} p_{ij}\log(p_{ij})$$

Higher values of these label quality scores correspond to tokens whose label is more likely to be correct. Let K denote the number of classes. For one sentence with n word-level tokens, we thus have:

- $\mathbf{p}$, a $n \times K$ matrix where p_{ij} is the model predicted probability that the ith token belongs to class j.
- $\mathbf{l} = [l_1, \dots, l_n]$, where l_i is the given class label of the ith token.
- $\mathbf{q} = [q_1, \dots, q_n]$, where q_i is the label quality of the ith token.
- $\mathbf{b} = [b_1, \dots, b_n]$, where $b_i = 1$ if the ith token is flagged as potentially mislabeled, otherwise $b_i = 0$.

Recall that to properly verify whether a token is really mislabeled, a reviewer must read the full sentence to understand the broader context. Thus the most efficient way to review labels in a dataset is to prioritize inspection of those sentences most likely to contain mislabeled token. We consider 11 methods to estimate an overall quality score $s(x)$ for the sentence, where higher values corresponds to sentences whose labels are more likely all correct.

1. ***predicted-difference***: the number of disagreements between the given and model-predicted class labels over the tokens in the sentence. Here we break ties in favor of the highest-confidence disagreement. More formally:
$$s(x) = -|\mathcal{R}| - \max_{i\in\mathcal{R}} p_{i,\widehat{l_i}}$$
where $\widehat{l_i} = \text{argmax}_j\{p_{ij}\}$ and $\mathcal{R} = \{i : \widehat{l_i} \neq l_i\}$. If $\mathcal{R} = \emptyset, \max_{i\in\mathcal{R}} p_{i,\widehat{l_i}} = 0$.
2. ***bad-token-counts***: $s(x) = -\sum_i b_i$, the number of Confident Learning flagged tokens.
3. ***bad-token-counts-avg***: again scoring based on number of tokens flagged as potentially mislabeled, but now breaking ties primarily via the average label quality score of the flagged tokens and secondarily via the average label quality score of the other tokens. More formally:
$$s(x) = -\sum_i b_i + \frac{1}{|\mathcal{R}|}\sum_{i\in\mathcal{R}} q_i + \frac{\epsilon}{|\mathcal{S}|}\sum_{i\in\mathcal{S}} q_i$$
where $\mathcal{R} = \{i : b_i = 1\}, \mathcal{S} = \{i : b_i = 0\}$, and ϵ is some small constant.
4. ***bad-token-counts-min***: similar to ***bad-token-counts-avg***, but break ties using minimum token quality rather than average token quality. More formally:
$$s(x) = -\sum_i b_i + \min_{i\in\mathcal{R}} q_i + \epsilon \cdot \min_{i\in\mathcal{S}} q_i$$
5. ***good-fraction***: fraction of tokens not flagged as potential issues:
$$s(x) = -\frac{1}{n}\sum_{i=1}^{n} b_i$$
6. ***penalize-bad-tokens***: penalize flagged tokens based on their corresponding label quality scores. More formally:
$$s(x) = 1 - \frac{1}{n}\sum_{i=1}^{n} b_i(1-q_i)$$
7. ***average-quality***: average label quality of tokens in the sentence: $s(x) = \frac{1}{n}\sum_{i=1}^{n} q_i$
8. ***product***: $s(x) = \sum_i \log(q_i + c)$, where c is a constant hyperparameter. This score places greater emphasis on token with low estimate label-quality, while still being influenced by all tokens' quality (like *average-quality*).
9. ***expected-bad***: a rough approximation of the expected number of mislabeled tokens in the sentence. More formally:
$$s(x) = \sum_{j=1}^{\min(n,J)} j \cdot q^{(j)}$$
where $q^{(i)}$ is the ith lowest token label quality score in the sentence, and J is a hyperparameter. If using the self-confidence label-quality score, $1 - q^{(i)}$ can be considered a loose proxy for the probability of having at least i label errors in this sentence.
10. ***expected-alt***: similar to *expected-bad*, but only considering the likelihood of any label error rather than how many might be in the sentence:
$$s(x) = \sum_{j=1}^{\min(n,J)} q^{(j)}$$
11. ***worst-token***: the quality of the worst-labeled token in the sentence determines its overall quality score, $s(x) = \min\{q_1, q_2, \dots, q_n\}$. This is a reasonable way to rank the sentences most likely to have some label error, ie. those most worthy of manual review.

➤Experiment

For evaluation, we apply each sentence scoring method to the given class labels in the CoNLL-2003, a widely used named-entity recognition dataset which contains 4 types of named entities: **PER** for persons, **ORG** for organizations, **LOC** for locations, **MISC** for miscellancous other entites, with **O** being reserved as a label for other types of words that are not named entities. The dataset is in IOB2 format, such that all named entities possess an extra *B-* or *I-* prefix, which indicates whether this token is the Beginning of an entity or an Intermediate part of one. We restrict our attention to the test set, for which all ground truth labels errors were identified by Wang et al in the **CoNLL++** dataset. We consider three different settings in our experiment: **bert-unmerged**: uses *bert-base-NER* to obtain a set of model-predicted probabilities on the 9 classes; **bert**: considers a fewer set of 5 classes by omitting the B- and I-prefixes to **focus more on severe error types**; **xlm**: obtains a **different set of model-predicted probabilities** using another pre-trained network *xlm-roberta-large-finetunes-conll03-english*, in which we again consider the reduced set of 5 classes. In either setting, **a sentence is considered mislabeled if at least one label of the word-level token differs from original dataset.**

➤Results

We consider evaluation metrics from information retrieval, which depend on the ranking of sentences induced by $s(x)$ rather than its magnitude. **Sentences that contain any mislabeled token are considered true positives** when we compute **AUROC** and **AUPRC**. Our third metric, **Lift @ #Errors**, measures how many times more prevalent labels errors are within the top-T scoring sentences vs. all sentences, where T is the number of true positives.

Sentence Score	Token Score	bert	xlm	bert-unmerged
`predicted-difference`		0.3422	0.3412	0.3190
`bad-token-counts`		0.3087	0.3186	0.3291
`bad-token-counts-avg`	`sc`	0.3740	0.3697	0.3768
	`nm`	0.3702	0.3603	0.3740
	`cwe`	0.3597	0.3597	0.3609
`bad-token-counts-min`	`sc`	0.3804	0.3759	0.3901
	`nm`	0.3744	0.3662	0.3822
	`cwe`	0.3695	0.3602	0.3607
`good-fraction`		0.3131	0.3159	0.2996
`average-quality`	`sc`	0.3022	0.3349	0.2574
	`nm`	0.3066	0.3143	0.2648
	`cwe`	0.2767	0.3495	0.2572
`penalize-bad-tokens`	`sc`	0.3423	0.3321	0.3229
	`nm`	0.3368	0.3126	0.3221
	`cwe`	0.3191	0.3380	0.3023
`product`	`sc`	0.3794	0.3559	0.3726
	`nm`	0.3807	0.3533	0.3823
	`cwe`	0.3519	0.3783	0.3359
`expected-bad`	`sc`	0.3383	0.3485	0.3532
	`nm`	0.3776	0.3227	0.3513
	`cwe`	0.3191	0.3541	0.2980
`expected-alt`	`sc`	0.3927	0.3628	0.3614
	`nm`	0.3850	0.3342	0.3603
	`cwe`	0.3335	0.3620	0.3114
`worst-token`	`sc`	**0.4357**	**0.4021**	**0.4236**
	`nm`	0.4243	0.3963	0.3933
	`cwe`	0.3215	0.3815	0.2974

The table above presents the **AUPRC** achieved by different sentence scoring (SC) and token scoring (TS) methods. Note that the token score field is left empty for sentence scoring methods that do not reply on token scores. Our results show that ***worst-token*** (using the *self-confidence* token-score) generally achieves the best LED performance across the three experiments. To most usefully rank sentences for identifying label errors, one should thus account for classifier confidence but not be directly influenced by all tokens' estimated quality, which may be noisy.

* Code implementation: https://github.com/cleanlab/token-label-error-benchmarks