Identifying Incorrect Labels in the CoNLL-2003 Corpus

Frederick Reiss[1,2], **Hong Xu**[2]*, **Bryan Cutler**[2]*,
Karthik Muthuraman[2]* and **Zachary Eichenberger**[1,3]*
[1]IBM Research – Almaden, San Jose, CA 95120, USA
[2]IBM Center for Open Source Data and AI Technologies (CODAIT),
San Francisco, CA 94105, USA
[3]University of Michigan, Ann Arbor, MI 48109, USA
frreiss@us.ibm.com, hongx@ibm.com, bjcutler@us.ibm.com
karthik.muthuraman@ibm.com, zachary.eichen@gmail.com

Abstract

The CoNLL-2003 corpus for English-language named entity recognition (NER) is one of the most influential corpora for NER model research. A large number of publications, including many landmark works, have used this corpus as a source of ground truth for NER tasks. In this paper, we examine this corpus and identify over 1300 incorrect labels (out of 35089 in the corpus). In particular, the number of incorrect labels in the test fold is comparable to the number of errors that state-of-the-art models make when running inference over this corpus.

We describe the process by which we identified these incorrect labels, using novel variants of techniques from semi-supervised learning. We also summarize the types of errors that we found, and we revisit several recent results in NER in light of the corrected data. Finally, we show experimentally that our corrections to the corpus have a positive impact on three state-of-the-art models.

1 Introduction

The English-language portion of the CoNLL-2003 shared task (Tjong Kim Sang and De Meulder, 2003) (henceforth CoNLL-2003) is one of the most widely-used benchmarks for named entity recognition (NER) models. It consists of news articles from the Reuters RCV1 corpus (Lewis et al., 2004).

Since its debut, CoNLL-2003 has played a central role in NLP research. Over 2300 research papers have cited the original CoNLL-2003 paper[1]. Among these works, many are landmark results that have revolutionized the field of natural language processing, including Glove embeddings (Pennington et al., 2014), BERT embeddings (Devlin et al., 2019), conditional random fields (Sutton and McCallum, 2012), and bidirectional LSTM models (Lample et al., 2016).

The CoNLL-2003 corpus continues to be used in NER research. The Papers with Code website (Paper with Code, 2020), which tracks state-of-the-art F1 scores[2] for this corpus, currently (as of July 2020) shows 43 results from 2016 through 2019 that improved this metric.

While researchers have relied heavily on the CoNLL-2003 corpus as a source of ground truth, few have paid attention to the corpus itself. Errors in the corpus could potentially mislead and even divert the course of future research. Recent work has pointed out that improper benchmarking can have significant impact on evaluating machine learning algorithms (Smith-Miles et al., 2014). The fact that Stanislawek et al. (2019) and Wang et al. (2019) found many errors while examining parts of the corpus is even more alarming. A detailed examination of the corpus has become imperative.

In this paper, we present our work on correcting labeling errors in the CoNLL-2003 corpus. Section 2 gives an overview of the corpus itself, the high-level process we followed, and related work. Section 3 describes how we used a novel form of semi-supervised labeling to identify potentially-incorrect labels. Sections 4 and 5 describe how we examined and categorized the flagged labels. And Sections 6 and 7 describe how we created a corrected version of the corpus and reevaluated past results.

2 Overview

The CoNLL-2003 corpus contains news articles from a subset of the Reuters RCV1 corpus (Lewis et al., 2004). Entities are tagged using an extended version of the tagging policy from the Message Understanding Conference (Tjong Kim Sang and

*The last four authors have contributed equally.

[1] https://scholar.google.com/scholar?cites=17103810098319730115

[2] "F1 score" here means "harmonic mean of precision and recall over the test fold for models trained on the train fold".

https://doi.org/10.18653/v1/P17

De Meulder, 2003) (MUC), with the addition of a new tag MISC to cover entities not mentioned in MUC's labeling rules. The data consists of text files in which each line holds information about one token. Associated with each token are tags in inside-outside-begin (IOB) format (Ramshaw and Marcus, 1995). The files, eng.train, eng.testa, and eng.testb, contain the train , dev , and test folds of the corpus, respectively.

2.1 Our Work

In this paper, we identify and correct labeling errors in the CoNLL-2003 corpus. We used a semi-supervised approach to flag potentially-incorrect labels in the corpus, then manually reviewed the labels thus flagged.

Our approach builds on previous work in semi-supervised labeling, with some key differences. Because we were looking for incorrect labels in a corpus that already had many high-quality labels, we needed a sieve with especially high sensitivity. We used ensembles of NER models trained on the corpus, and we focused on cases where the models agreed strongly on a particular label, but that label does not appear in the corpus. One of these ensembles was the outputs of the original 16 entries in the 2003 competition. We also trained two other 17-model ensembles ourselves by applying Gaussian random projections to the BERT embeddings space.

We deliberately used models with F1 scores significantly below the state of the art. To find incorrect labels, we needed models that disagree with the original CoNLL-2003 corpus. Our initial experiments with the CoNLL-2003 competition entries showed that this ensemble, with F1 scores between 0.6 and 0.88, was particularly effective for finding incorrect labels. We tuned the models that we trained ourselves to have F1 scores in this range.

Our technique flagged 3182 out of a total of 35089 entity labels. Manual inspection determined that 850 of these labels — 27% — were incorrect. We also found 470 additional incorrect labels in close proximity to the labels that our techniques flagged, for a total of 1320 incorrect labels across the corpus.

Of a particular note, our analysis found 421 incorrect labels in the test fold. The test fold for this corpus contains 5648 labels. An F1 score of 0.93, as current state-of-the-art models produce, corresponds to approximately 400 errors on this fold. The change in F1 score over the past 17 years (0.934 - 0.888 = 0.046) corresponds to eliminating approximately 300 errors. The error rate of state-of-the-art models is comparable to the error rate of the corpus itself.

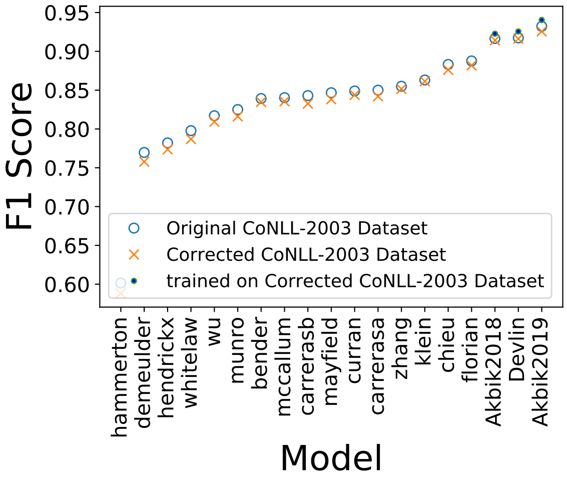


Figure 1: F1 scores on the test fold for 18 different NER models before and after correcting labeling errors in the test fold. Correcting these errors widened the spread in F1 scores between the more sophisticated models at the right and the less sophisticated models at the left.

We used the results of our hand labeling to build a corrected version of the corpus. Then we re-evaluated the original entries in the competition, plus selected NER models from recent work, over the corrected corpus.

Figure 1 shows how the reported accuracy of these models changed. Surprisingly, we did not observe any change in the relative ranking of the models. Even though we corrected almost 8% of the labels in the test fold, no model's F1 score changed by more than 0.01. Without retraining, the changes in F1 score were all in the downward direction, but the F1 scores of the more sophisticated models dropped by less. The highest F1 score dropped from 0.932 to 0.927, while the lowest dropped from 0.601 to 0.589. When we retrained the three state-of-the-art models on the corrected data, their F1 scores became higher than their original scores.

2.1.1 Reproducibility

We have shared the full data set for this paper at https://github.com/CODAIT/Identifying-Incorrect-Labels-In-CoNLL-2003. This data set includes a complete list of the errors that we found in the corpus, with notes from the labelers about the nature of each error. We also include scripts for generating a corrected version of the full CoNLL-2003 corpus in its original format. We have also released the code for our experiments as part of our open source *Text Extensions for Pandas* project[3].

2.2 Related Work

Most of the previous work we have mentioned so far has treated the CoNLL-2003 corpus as ground truth. Two recent exceptions to this trend are Stanislawek et al. (2019) and Wang et al. (2019).

Stanislawek et al. (2019) identified some of the same incorrect labels that we found. This paper categorized the errors that modern NER models make on the `test` fold of the corpus. As a side-effect of the error analysis, the authors of this paper flagged cases where the output of a model had been considered "wrong" because a label in the corpus was incorrect. The authors identified 99 such errors in the `test` fold of this corpus.

There are several important differences between this paper and our work. Stanislawek et al. (2019) flagged errors as a side-effect of another task, while our primary focus was on identifying as many errors as possible. Due to our broader focus, we identified 421 errors in the `test` fold, compared to the 99 errors they found. We also examined the other two folds of the corpus, while the previous paper focused only on the `test` fold. The previous paper used models with high precision and recall; and they examined all the incorrect outputs of these models. We deliberately used less accurate models so as to widen the scope of potential errors flagged, and we focused on cases where there was strong agreement between these models plus disagreement with the ground truth data.

Wang et al. (2019) hired human labelers to label all sentences in the `test` fold of the corpus and found that 5.38% of sentences in this fold contained errors. This number is a lower error rate than we report, mostly due to the fact that the labelers did not look for errors in tokenization or sentence identification. Excluding those types of errors, our work flagged 348 out of 5648 entities in the `test` fold, for an error rate of 6.16%. We attribute the remaining 0.78 percent increase in error rate to the fact that our labelers examined entire documents and looked for consistency across documents, while Wang et al. (2019)'s labelers only viewed individual sentences in isolation.

Beyond the larger number of types of errors we searched for, there are two other important differences between our work and that of Wang et al. (2019). We developed a novel semi-supervised approach to identifying incorrect labels, and we used this approach to examine the entire corpus instead of just the `test` fold.

Our general approach of training an ensemble of models, then focusing attention on areas where most of the models disagree with the existing labels, has parallels to other work on human-in-the-loop methods for creating ground truth. Liang et al. (2017) used confidence estimates from a model trained on a data set to flag potential errors in the same data set for further review. The specific NLP task studied in that work was that of extracting a list of patient problems from an electronic medical record.

Fusing together the output of multiple models and/or rules is a also common approach when using weak supervision to train models over unlabeled NLP corpora. Lison et al. (2020) used hidden Markov models to generate labeled NER data from the outputs of multiple labeling functions. The Snorkel system (Ratner et al., 2020) provides a general framework for using the outputs of labeling functions to estimate both labels and the confidence of those labels.

The data management and data mining communities have a long history of building systems and algorithms to identify errors in ground truth data. Abedjan et al. (2016) provide a through survey. Although the primary focus of this previous work was on structured data, subtasks like address normalization have an NLP component.

3 Automated Labeling

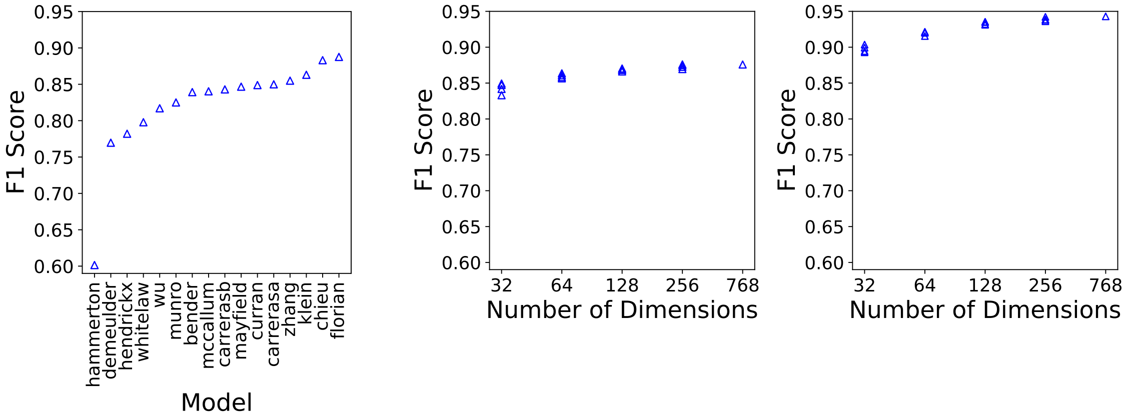
We did not set out to relabel the CoNLL-2003 corpus. When we started looking at this corpus, our intent was to identify entity mentions that older models are not able to extract, but that state-of-the-art models are able to extract. We had hoped to use this information to drive continued improvements to these models.

3.1 Initial Results

The downloadable archive[4] for the corpus includes the outputs from the original entrants in the 2003

[3] https://github.com/CODAIT/text-extensions-for-pandas

[4] https://www.clips.uantwerpen.be/conll2003/ner/



(a) Original CoNLL-2003 entries.

(b) BERT-based models.

(c) BERT + cross-validation.

Figure 2: F1 scores of the models in our three ensembles. Each scatter point in these plots represents a trained model. The x-axes of Figures 2b and 2c represent the number of dimensions of embeddings.

competition. These entrants used a variety of different models, drawing on the technology available at the time.

We believed that these model outputs would provide an objective picture of what kinds of entities were difficult to extract for state-of-the-art models circa 2003. We hypothesized that there would be entity mentions that none of the models could extract correctly, due to limitations of 2003-era technology. We further believed that modern models would be able to tag some of these previously impossible mentions. To test this hypothesis, we aggregated together the outputs of the original entrants to find these "difficult" entities.

The corpus ships as a collection of tokens with tags in IOB format. Using *Text Extensions for Pandas*[5], a library of extension types for `pandas` DataFrames (Reback et al., 2020; McKinney et al., 2010), we translated the labeled tokens of the corpus into *entity mentions* — that is, *spans* of tokens within the corpus's document, plus the corresponding entity type *tag* for each span.

We performed the same translation on each of the entrants' outputs. This process produced seventeen sets of entity mentions: One for the original corpus and one for each of the sixteen entrants. Next, we merged these sets together to find the mentions that were present in the original corpus but were *not* present in the competition entries.

Then we looked at some of these entity mentions in the context of the original news articles, and our original hypothesis fell apart. About one third the examples we looked at turned out to be incorrect labels. It would be hard to argue that these "incorrect" answers were due to inadequacies of early-2000's technology, when it was in fact the *corpus* that was incorrect.

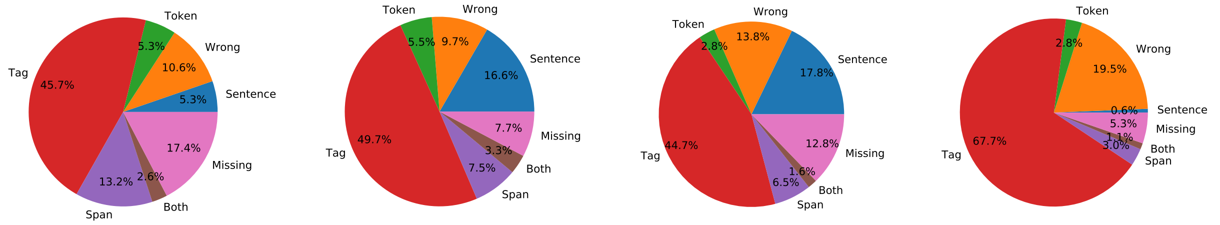
Next, we took a slightly different view of the aggregate data we had. Instead of looking for entity mentions that were in the corpus but not in the entrants' outputs, we looked for entity mentions that were in all the entrants' outputs but were *not* in the corpus. As before, a third of the examples that we looked at involved incorrect or missing labels. We decided at this point to focus on identifying and correcting these incorrect labels.

3.2 Training Custom Models

The model outputs from the original CoNLL-2003 competition had proven useful for zeroing in on incorrect labels, but this data had a significant shortcoming. The model outputs only cover the `dev` and `test` folds of the corpus. No model outputs on the `train` fold are available. To apply the technique we had used so far to the `train` fold, we would need to train our own collection of models.

We used a BERT embeddings layer from the `transformers` open source library (Wolf et al., 2019), tuned on the CoNLL-2003 corpus, to produce BERT embeddings over sliding windows of text from the `train` fold. Then we applied 16 different Gaussian random projections to these 768-dimensional embeddings to reduce them to between 32 and 256 dimensions. We trained multinomial logistic regression classifiers over these random projections. We also trained an additional classifier over the full embeddings, for a total of 17 different models.

[5] https://github.com/CODAIT/text-extensions-for-pandas



(a) Percentage of each error type correctly flagged by using original entrants.

(b) Percentage of each error type correctly flagged by the method using custom models.

(c) Percentage of each error type correctly flagged by the method using cross-validation.

(d) Percentage of Each Error Type Correctly Flagged By hand labeling.

Figure 3: Distribution of error types found by each of the four methods.

Our goal in building these models was not to attain the highest possible precision and recall. In fact, high levels of accuracy could be detrimental to our task, as high levels of accuracy imply a high congruence with the ground-truth labels we were trying to correct. Instead, we wanted a collection of models that would produce diverse results and F1 scores in line with the accuracy of the original CoNLL-2003 entrants.

With one exception, the CoNLL-2003 competition produced F1 scores between 77% and 89% on the `test` fold. We tuned our models' training and inference until they produced results approximately within this range. Figure 2 shows the resulting F1 scores on the `test` fold. Figure 2a shows the original CoNLL-2003 competition entries' F1 scores, while Figure 2b shows the F1 scores of the models we trained, plotted against the dimensionality of their Gaussian random projection stages.

Based on published results on BERT embeddings for NER, we expect that additional tuning would have raised the F1 scores of our models by about 0.02. We judged that the lower F1 scores in Figure 2 are better for this application.

As with our previous analysis of the original competition results, we aggregated together the outputs of these 17 models on the `test` fold of the corpus, then aligned these results with the corpus labels. A manual spot-check of these aligned results verified that these aggregated results also functioned as an effective sieve for identifying incorrect labels. Roughly half of the entity mentions that were found by all 17 models but were not in the corpus were due to incorrect or missing labels in the corpus. We found similar results on the `dev` fold.

3.3 Cross-Validation

Next, we applied our ensemble of models to the `train` fold of the corpus and compared the aggregated results against the corpus's labels. As with the `test` fold, we were able to use the aggregate model outputs to identify a list of entity mentions with a high fraction of incorrect corpus labels. However, this list was significantly shorter than the lists we were able to produce on the `test` and `dev` folds. Because the models were themselves trained on the `train` fold, there were fewer discrepancies between the model outputs and the corpus labels.

To produce a larger list of potentially incorrect labels, we divided the entire corpus randomly into ten folds and performed a ten-fold cross-validation. For each of the ten folds, we retrained our ensemble of models on the other 9 folds and ran model inference on the current fold. This process involved training 170 different models, but because we only needed to generate the BERT embeddings once, we were able to perform all training in a few hours on a 4-year-old MacBook.

Interestingly, this cross-validation approach produced models with significantly higher F1 scores on the random holdout sets, compared with our earlier approach of training on the `train` fold and testing on the `test` fold. As Figure 2c shows, F1 scores for the holdout sets for each of the models — which together encompass the entire corpus — ranged from 0.89 to 0.94, an increase of roughly 5%.

We attribute this improvement to the non-random split of the original corpus. The contest judges used article publication date to split the corpus into folds. The `train` and `dev` folds used articles from August of 1996, while the `test` fold was from December of that year (Tjong Kim Sang and De Meulder, 2003).

This non-random split matches common industry practices[6]. However, dividing the the corpus by time means that any systematic changes in the target domain over time are not visible to the optimizer during training. Models trained on a random sample of the corpus are able to achieve a higher F1 score because they have better information about the types of articles that were published in December 1996.

In both ensembles that we trained, our model outputs aligned well with the labels on the `train` fold. Consequently, our sieve identified fewer potentially incorrect labels in the `train` fold of the corpus, which in turn would lead to our identifying fewer incorrect labels during manual relabeling. Better accuracy led to worse results.

4 Hand Labeling

Each of our three ensembles produced two lists of labels: one list of labels that were in the corpus but not in the model outputs; and a second list of labels that were in the model outputs but not in the corpus. Overall, we produced six lists of potentially-incorrect labels. Four of these lists spanned the entire corpus, while the remaining two (from the original contest entries) only spanned the `test` and `dev` folds.

We proceeded to examine these lists by hand, looking at each flagged label in the context of the target document. We focused on the labels where there was a strong agreement between the models in each ensemble. We started out by examining the labels where all models agreed, then moved onto the labels where all models but one agreed, and so on. As we progressed to labels with less agreement among models, the fraction of flagged labels that was actually incorrect decreased. When this fraction dropped below 20 percent, we stopped going through the ordered list of flagged labels.

For each list of potentially-incorrect labels, one member of our team examined the labels, and a second member of our team audited the decisions that the first member had made. In total, we made 12 passes (3 ensembles × 2 sets of labels × 2 human reviewers) of manual review over the `train` and `test` folds of the corpus and 8 passes over the `test` fold.

When we found that a label was incorrect, we coded the type of error and the required correction so that the error could be corrected automatically later on. We divided errors into several categories:

- `Tag`: The corpus correctly identifies the span of an entity mention, but the span is associated with the wrong entity type.
- `Span`: The corpus correctly identifies the type of an entity mention, but the boundaries of the span of tokens containing the mention are incorrect.
- `Both`: The corpus correctly identifies an entity mention, but both the tag and the span boundaries are incorrect.
- `Wrong`: The corpus incorrectly identifies an entity mention.
- `Sentence`: The corpus contains an incorrect sentence boundary, and as a result the span and/or tag of one or more entity mentions are incorrect. This type of error especially problematic because incorrect labels on both sides of the sentence boundary count as two mistakes when computing precision and recall.
- `Token`: The corpus contains an incorrect token boundary, and as a result the span and/or tag of one or more entity mentions are incorrect.

Appendix 9.1 shows examples of each error type. The data set that we have published as a companion to this paper (See Section 2.1.1) includes complete lists of the errors that we found, both before and after manual review.

4.1 Inter-Annotator Agreement

Each manual pass over the corpus involved validating a set of suggested changes, not reannotating the corpus in its entirety. As a result, conventional metrics of inter-annotator agreement between our human evaluators do not apply. Instead, we report the similarity between the outputs of the three *ensembles*.

Table 1 summarizes the Jaccard similarity between the three ensembles' outputs before and after manual review. Figure 4 shows a Venn diagram view of the relationship between the sets of flagged labels after manual review. The raw outputs of the two BERT-based ensembles showed a high degree

[6] In our experience, most organizations that use machine learning do not have the ability to travel backwards in time. Hence, they train models on data from the past and apply those models to data from the future.

Ensemble 1	Ensemble 2	Fold(s)	Before Review	After First Review	After Second Review
Original models	Custom models	dev / test	0.5153	0.2500	0.2533
Original models	Custom + Cross-val.	dev / test	0.5179	0.2072	0.2052
Custom models	Custom + Cross-val.	dev / test	0.8220	0.5167	0.5532
Custom models	Custom + Cross-val.	train	0.8707	0.6677	0.6592

Table 1: Jaccard similarity between the flagged labels from different pairs of ensembles before and after human review. The original models flagged a substantially different set of labels from our BERT-based custom models, and this divergence increased after manual review.

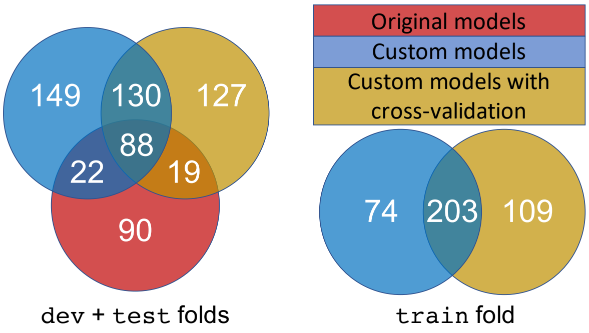


Figure 4: Number of errors flagged by different combinations of ensembles after filtering by human labelers.

of overlap, but this overlap reduced substantially after manual review. The original models flagged a very different set of labels from the BERT based models, especially after manual review.

5 Incorrect Labels Identified

In total, we examined 3182 labels our ensembles had flagged in the three folds of the corpus. We considered any label where fewer than 7 models agreed with the corpus label to be "flagged". Of these labels, 1274 came from the test fold, 854 came from the dev fold, and 1054 came from the train fold; accounting for 22.6%, 14.3%, and 4.5% of their folds, respectively.

As we noted in Section 3.3, our models had significantly higher F1 scores on the train fold, both with and without cross-validation. Because model outputs were closer to the corpus labels, the ensembles flagged fewer labels on this fold. However, the fraction of these labels that were actually incorrect was higher than that on the other folds: 34% versus 23%.

Of the errors correctly flagged, 184 were found by the ensemble composed of the original entrants' results; 641 were flagged by our custom models; and 275 errors were found by custom models with cross-validation. 372 of these errors were correctly flagged by two or more approaches. While examining the affected documents, we found 470 additional errors in the vicinity of flagged errors.

Figure 3 shows the distribution of errors broken down by error type and source. The most frequent error type we found in the corpus was a Tag type error, accounting for 48% of errors in total. Both type errors were least frequent.

Our BERT-based models found a higher fraction of Sentence type errors, largely because these models were able to express spans that cross sentence boundaries. The entrants' outputs in our first ensemble, being constrained by the IOB file format, were physically incapable of expressing a span that crosses a sentence boundary. We also suspect that many of these older models operated on one sentence at a time, while the document context feeding our BERT embeddings could span multiple sentences.

Had we been aiming to maximize the F1 scores of our BERT-based models, we would have post-processed the outputs of these models to split spans along sentence boundaries. This lack of post-processing led to a decrease in F1 score, but it enabled us to find more errors.

The distribution of error types remained relatively constant across folds, with one exception: Sentence errors accounted for a much larger fraction in the train fold — 26% of errors, as opposed to the 8% and 10% rates in the dev and test folds, respectively.

6 Corrected CoNLL-2003 Corpus

After identifying incorrect tags, spans and sentence boundaries, we created a corrected version of the original CoNLL-2003 corpus, which we refer to as the corrected CoNLL-2003 corpus.

We used the *Text Extensions for Pandas* library to parse the original corpus and extract tokens and spans for each entity. We created data files containing all of the vetted corrections from our hand labeling of ensemble outputs. We wrote a script that applies all of these corrections to the CoNLL-2003 corpus, producing a corrected version of the

Entrant	**Original** test Fold			**Corrected** test Fold		
	Precision	Recall	F1 Score	Precision	Recall	F1 Score
bender	0.8468	0.8318	0.8392	0.8412	0.8279	0.8347
carrerasa	0.8405	0.8596	0.8500	0.8325	0.8517	0.8420
carrerasb	0.8581	0.8284	0.8430	0.8466	0.8192	0.8327
chieu	0.8812	0.8851	0.8831	0.8733	0.8791	0.8762
curran	0.8429	0.8550	0.8489	0.8376	0.8501	0.8438
demeulder	0.7584	0.7813	0.7697	0.7468	0.7693	0.7579
florian	0.8899	0.8854	0.8876	0.8837	0.8800	0.8818
hammerton	0.6909	0.5326	0.6015	0.6749	0.5210	0.5881
hendrickx	0.7633	0.8017	0.7820	0.7548	0.7936	0.7737
klein	0.8612	0.8649	0.8631	0.8593	0.8634	0.8614
mayfield	0.8445	0.8490	0.8467	0.8354	0.8411	0.8382
mccallum	0.8452	0.8355	0.8404	0.8398	0.8315	0.8356
munro	0.8087	0.8421	0.8251	0.8003	0.8327	0.8162
whitelaw	0.8160	0.7805	0.7978	0.8045	0.7702	0.7870
wu	0.8202	0.8139	0.8170	0.8112	0.8075	0.8094
zhang	0.8613	0.8488	0.8550	0.8574	0.8459	0.8516

Table 2: Experimental results on the original CoNLL 2003 (English) competition.

corpus.

For information on how to obtain the code and data necessary to recreate our corrected corpus, as well as all the experiment code for this paper, see Section 2.1.1.

7 Experimental Evaluation

In this section, we first re-evaluate the entries from the original competition against the corrected test fold of the corpus. We then re-evaluate the metrics of three state-of-the-art NER models from recent literature on the corrected corpus.

7.1 Re-evaluation of the Original Competition Entries

We evaluated the original 16 CoNLL-2003 competition entries on the original and corrected CoNLL-2003 test folds.

Before evaluating on the corrected data, we needed to adjust sentence boundaries and tokenization in the entrants' output files to match that of the corrected corpus. The evaluation metric for this corpus relies on perfect alignment between tokens and sentences of the files being compared. When we split a token, we copied the token's label to the new, smaller tokens.

We recomputed precision, recall, and F1 scores. Our results are shown in Table 2 and Figure 1. All of the entries have lower precision, recall, and F1 scores on the corrected CoNLL-2003 test fold than on the original test fold. Although we changed nearly 8% of the labels in the test fold, all the models' metrics decreased by 1% or less.

The more accurate entries saw their F1 scores decline by less than the entries with lower F1 scores. For example, the top-scoring entry's F1 score dropped by 0.0054, while the bottom-scoring entry dropped by 0.0122 — more than twice as much. As a result, the ranking of entries did not change. It appears that the errors in the original corpus penalize models that produce answers closer to the actual ground truth.

Since we did not have access to the original models, we only performed inference and scoring on the corrected CoNLL-2003 corpus. We expect that the metrics would improve if the models are entirely re-trained on the corrected corpus' train fold. This would constitute relevant future work and point towards new reliable benchmarks.

7.2 Experimental Results on Recent Models

We evaluated three state-of-the-art NER models. We selected three models (Akbik et al., 2018, 2019; Devlin et al., 2019) according to the ranking of models on the CoNLL-2003 NER task compiled on Papers with Code (Paper with Code, 2020)[7]. Table 3 summarizes our experimental results. We have the following observations.

[7] We initially planned to select all of the models that rank top 10 from (Paper with Code, 2020). However, we were able to reproduce only three of them. We were unable to apply the rest of the models for the following technical reasons: two of which we requested code from the authors never received any responses; one of which we could find code but there is no instruction on how to use the code; three of which we could find code with instructions but we could not reproduce by following the instructions; one of which uses a nonstandard tagging scheme. We have contacted the authors of all of these papers for help with their code.

Model		**Original** test Fold			**Corrected** test Fold		
		Precision	Recall	F1 Score	Precision	Recall	F1 Score
Trained on **Original** Corpus	(Akbik et al., 2018)	0.9133	0.9207	0.9165	0.9108	0.9177	0.9142
	(Akbik et al., 2019)	0.9290	0.9354	0.9322	0.9226	0.9286	0.9256
	(Devlin et al., 2019)	0.9119	0.9229	0.9173	0.9110	0.9217	0.9163
Trained on **Corrected** Corpus	(Akbik et al., 2018)	0.9073	0.9120	0.9096	0.9206	0.9248	0.9227
	(Akbik et al., 2019)	0.9252	0.9260	0.9256	0.9400	0.9407	0.9404
	(Devlin et al., 2019)	0.9228	0.9309	0.9268	0.9218	0.9295	0.9256

Table 3: Experimental results for recent models. We trained each of the three models (Akbik et al., 2018), (Akbik et al., 2019), and (Devlin et al., 2019) on the original and corrected train folds, respectively. For each trained model, we evaluated on the original and corrected test folds, respectively. For (Akbik et al., 2018) and (Akbik et al., 2019), we trained on both train and dev folds. For (Devlin et al., 2019), we trained on the train fold. For all models, we used the hyperparameter settings specified in their respective papers.

On the corrected test fold, all the listed metrics (the F1 scores, precision, and recall) are higher for the models trained on the corrected corpus than those on the original corpus. This indicates that our correction on the corpus has a positive impact on the quality of training of the three models.

Comparing the metrics of models trained and evaluated on the original corpus (the top-left section of the table) and the metrics of models trained and evaluated on the corrected corpus (the bottom-right section of the table), we see that all metrics have been improved on the corrected corpus. This might indicate that these three models are actually more effective (according to the evaluation on the corrected corpus) than they were thought to be (according to the evaluation on the original corpus).

However, on the original test fold, all the listed metrics (the F1 scores, precision, and recall) are not higher for the models trained on the original corpus than those on the corrected corpus. This might be explained by the fact that the errors in the original test fold are not consistent with the original train and dev folds, hence models trained on the original corpus are not necessarily more advantageous than those trained on the corrected corpus when evaluated on the original test fold.

For models trained on the original corpus, all the listed metrics (the F1 scores, precision, and recall) on the corrected test fold are very close to those on the original test fold (differences are mostly within 0.002 and no larger than 0.01). Once again, this might be explicable by the fact that the errors the errors in the original test fold are not consistent with the original train and dev folds. Hence, models trained on the original corpus are not necessarily more advantageous when evaluated on the original test fold than on the corrected test fold.

8 Conclusion and Future Work

The CoNLL-2003 corpus is highly influential in named entity recognition (NER) research. It has been used for benchmarking many landmark NER models and has been continuing to play a critical role in recent research. In this paper, we took a closer look at the CoNLL-2003 corpus and identified a number of errors. We used a semi-supervised method to identify these errors and then systematically corrected them.

The primary contribution of this paper is the creation of a more error-free version of the CoNLL-2003 corpus, which can potentially be used to evaluate past NER models more accurately and make future benchmarking more reliable. Indeed, as our experiments on three recent state-of-the-art NER models have shown, our corrections to the corpus have a positive impact on these models: When evaluated on our corrected test fold, all three models trained on our corrected corpus outperformed their counterparts trained on the original corpus by a non-negligible margin.

We firmly believe that benchmarking corpora are the lighthouses for research, and improving the quality of benchmarking corpora is of utmost importance in guiding the research community. We hope that others can replicate the process we applied to this corpus on other key corpora, and in doing so, improve the utility of these vital resources.

References

Ziawasch Abedjan, Xu Chu, Dong Deng, Raul Castro Fernandez, Ihab F. Ilyas, Mourad Ouzzani, Paolo Papotti, Michael Stonebraker, and Nan Tang. 2016. Detecting data errors: Where are we and what needs to be done? *Proceedings of the VLDB Endowment*, 9(12):993–1004.

Alan Akbik, Tanja Bergmann, and Roland Vollgraf. 2019. Pooled contextualized embeddings for named entity recognition. In *Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 724–728.

Alan Akbik, Duncan Blythe, and Roland Vollgraf. 2018. Contextual string embeddings for sequence labeling. In *Proceedings of the International Conference on Computational Linguistics*, pages 1638–1649.

Walter Daelemans, Jakub Zavrel, A. van den Bosch, and Ko van der Sloot. 2002. MBT: Memory based tagger, version 1.0, reference guide. Technical report, University of Antwerp.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 4171–4186.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. In *Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

David D. Lewis, Yiming Yang, Tony G. Rose, and Fan Li. 2004. Rcv1: A new benchmark collection for text categorization research. *Journal of Machine Learning Research*, 5:361–397.

Jennifer J. Liang, Ching-Huei Tsou, and Murthy V. Devarakonda. 2017. Ground truth creation for complex clinical nlp tasks – an iterative vetting approach and lessons learned. *AMIA Summits on Translational Science Proceedings*, 2017:203 – 212.

Pierre Lison, Jeremy Barnes, Aliaksandr Hubin, and Samia Touileb. 2020. Named entity recognition without labelled data: A weak supervision approach. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 1518–1533.

Wes McKinney et al. 2010. Data structures for statistical computing in Python. In *Proceedings of the 9th Python in Science Conference*, volume 445, pages 51–56.

Paper with Code. 2020. Named Entity Recognition on CoNLL 2003 (English). `https://paperswithcode.com/sota/named-entity-recognition-ner-on-conll-2003`.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Lance Ramshaw and Mitch Marcus. 1995. Text chunking using transformation-based learning. In *Proceedings of the third Workshop on Very Large Corpora*.

Alexander Ratner, Stephen H. Bach, Henry R. Ehrenberg, Jason Alan Fries, Sen Wu, and Christopher Ré. 2020. Snorkel: Rapid training data creation with weak supervision. *The VLDB Journal*, 29:709–730.

Jeff Reback, Wes McKinney, jbrockmendel, Joris Van den Bossche, Tom Augspurger, Phillip Cloud, gfyoung, Sinhrks, Adam Klein, Simon Hawkins, and et al. 2020. pandas-dev/pandas: Pandas 1.0.5.

Kate Smith-Miles, Davaatseren Baatar, Brendan Wreford, and Rhyd Lewis. 2014. Towards objective measures of algorithm performance across instance space. *Computers & Operations Research*, 45:12–24.

Tomasz Stanislawek, Anna Wróblewska, Alicja Wójcicka, Daniel Ziembicki, and Przemyslaw Biecek. 2019. Named entity recognition - is there a glass ceiling? In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 624–633, Hong Kong, China. Association for Computational Linguistics.

Charles Sutton and Andrew McCallum. 2012. An introduction to conditional random fields. *Foundations and Trends in Machine Learning*, 4(4):267–373.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the SIGNLL Conference on Computational Natural Language Learning*, pages 142–147, USA. Association for Computational Linguistics.

Zihan Wang, Jingbo Shang, Liyuan Liu, Lihao Lu, Jiacheng Liu, and Jiawei Han. 2019. Crossweigh: Training named entity tagger from imperfect annotations. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 5153–5162. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2019. HuggingFace's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771.

9 Appendix

9.1 Types of Errors

We classified the errors that we found into several categories. In this section, we give concrete examples of each type.

9.1.1 `Tag` Errors

In some cases, the corpus had correctly identified the span of the entity mention, but the tokens of that span were labeled with an incorrect entity type. For example, the 156th document in the `test` fold contains the token/label sequence:

```
smuggled O
heroin O
from O
Turkey I-LOC
to O
Antwerp I-ORG
```

This sequence incorrectly tags a mention of the city `Antwerp` as an `ORG` entity when it should be tagged `LOC`.

We call errors of this type `Tag` errors.

9.1.2 `Span` Errors

In other cases, the corpus correctly identified the entity type of an entity mention, but there was an error in labeling the precise range of tokens containing that entity. For example, the 113th document of the `test` fold contains the token/label sequence:

```
Ingeborg I-PER
Helen I-PER
Markein O
```

This sequence incorrectly marks the span 'Ingeborg Helen' as a 'PER' entity, when the correct span is 'Ingeborg Helen Markein', the full name of a Norwegian skier.

We call errors of this type `Span` errors.

9.1.3 `Both` Errors

At some locations in the corpus, an entity was subject to both a `Span` error and a `Tag` error at the same time. For example, the headline for the 23rd document of the `test` fold contains the token/label sequence:

```
ARAB I-MISC
CONTRACTORS O
WIN O
AFRICAN I-MISC
CUP I-MISC
```

These labels miss an instance of the `ORG` entity `ARAB CONTRACTORS`, a reference to The Arab Contractors Sporting Club, an Egyptian soccer team. In lieu of labeling `ARAB CONTRACTORS`, the sequence labels `ARAB` as a single-token `MISC` entity, which is not correct because that token is part of the longer `ORG` entity.

We call errors of this type `Both` errors.

9.1.4 `Wrong` Errors

In some cases, the corpus marks tokens that do not match any entity type at all. For example, the 153rd document in the `test` fold contains the token/label sequence:

```
next O
Wednesday I-ORG
```

This sequence of labels incorrectly marks `Wednesday` as an `ORG` entity when that token is in fact a reference to a day of the week.

We call errors of this type `Wrong` errors.

9.1.5 `Sentence` Errors

The creators of the corpus used automatic tools to break each document into sentences. Some of these sentence boundaries were incorrect, and some of these incorrect sentence boundaries occurred in the middle of an entity mention. For example, the 20th document of the `dev` fold contains the token/label sequence:

```
the O
Berlin I-MISC

Grand I-MISC
Prix I-MISC
```

(where the blank line encodes a sentence boundary).

Because the labeling and scoring scheme for this corpus does not permit entity mentions to span sentence boundaries, this sequence marks `Berlin` and `Grand Prix` as two separate 'MISC' entities.

This type of error is especially problematic because incorrect labels on these tokens will count as two mistakes when computing precision and recall. In addition, many models process one sentence at a time. When processing the above document, such models will see a sentence that ends with the token `Berlin`, followed by a sentence that starts with `Grand Prix`.

In other cases, an incorrect sentence boundary led the human labeler to conclude incorrectly that the period after an abbreviation is not part of the abbreviation. For example, the 208th document of the `train` fold contains the token/label sequence:

```
The I-ORG
Walt I-ORG
Disney I-ORG
Co I-ORG
. O

said O
Thursday O
```

(where the blank line encodes a sentence boundary).

In this example, `Co.` should be labeled as an `ORG` entity, but only `Co` (without the period) is marked.

We call errors of both these types `Sentence` errors.

9.1.6 "Token"-Type Errors

The authors of the original corpus used the MBT tagger (Daelemans et al., 2002) to tokenize the original news articles. Occasionally, the tokenizer made a mistake; and occasionally, a tokenization mistake happened to coincide with an entity mention. For example, the 169th document of the `train` fold contains the token/label sequence:

```
Nigerian I-MISC
terms O
jeopardize O
Commonwealth I-ORG
trip-Canada I-MISC
. O
```

Here, the tokenizer has incorrectly tokenized "trip — Canada" as a single token, and the human labeler has labeled this token as `MISC`, even though `Canada` is a `LOC` entity. Correcting this kind of problem involves splitting the incorrect token into its corrected parts, then relabeling those parts as needed. The above example turns into:

```
Nigerian I-MISC
terms O
jeopardize O
Commonwealth I-ORG
trip O
- O
Canada I-LOC
. O
```

We call errors of this type `Token` errors.