

Validating Label Consistency in NER Data Annotation

Qingkai Zeng[†], Mengxia Yu[†], Wenhao Yu[†], Tianwen Jiang[‡],
Meng Jiang[†]

[†]University of Notre Dame, Notre Dame, IN, USA

[‡]Harbin Institute of Technology, Harbin, Heilongjiang, China

[†]{qzeng, myu2, wyu1, mjiang2}@nd.edu

[‡]{twjiang}@ir.hit.edu.cn

Abstract

Data annotation plays a crucial role in ensuring your named entity recognition (NER) projects are trained with the correct information to learn from. Producing the most accurate labels is a challenge due to the complexity involved with annotation. Label inconsistency between multiple subsets of data annotation (e.g., training set and test set, or multiple training subsets) is an indicator of label mistakes. In this work, we present an empirical method to explore the relationship between label (in-)consistency and NER model performance. It can be used to validate the label consistency (or catch the inconsistency) in multiple sets of NER data annotation. In experiments, our method identified the label inconsistency of test data in SCIERC and CoNLL03 datasets (with 26.7% and 5.4% label mistakes). It validated the consistency in the corrected version of both datasets.

1 Introduction

Named entity recognition (NER) is one of the foundations of many downstream tasks such as relation extraction, event detection, and knowledge graph construction. NER models require vast amounts of labeled data to learn and identify patterns that humans cannot continuously. It is really about getting accurate data to train the models. When end-to-end neural models achieve excellent performance on NER in various domains (Lample et al., 2016; Liu et al., 2018; Luan et al., 2018; Zeng et al., 2020, 2021), building useful and challenging NER benchmarks, such as CoNLL03, WNUT16, and SCIERC, contributes significantly to the research community.

Data annotation plays a crucial role in building benchmarks and ensuring NLP models are trained with the correct information to learn from (Luan et al., 2018; Jiang et al., 2020; Yu et al., 2020). Producing the necessary annotation from any asset at

scale is a challenge, mainly because of the complexity involved with annotation. Getting the most accurate labels demands time and expertise.

Label mistakes can hardly be avoided, especially when the labeling process splits the data into multiple sets for distributed annotation. The mistakes cause label inconsistency between subsets of annotated data (e.g., training set and test set or multiple training subsets). For example, in the CoNLL03 dataset (Sang and De Meulder, 2003), a standard NER benchmark that has been cited over 2,300 times, label mistakes were found in 5.38% of the test set (Wang et al., 2019). Note that the state-of-the-art results on CoNLL03 have achieved an F1 score of $\sim .93$. So even if the label mistakes make up a tiny part, they cannot be negligible when researchers are trying to improve the results further. In the work of Wang *et al.*, five annotators were recruited to correct the label mistakes. Compared to the original test set results, the corrected test set results are more accurate and stable.

However, two critical issues were not resolved in this process: *i*) How to identify label inconsistency between the subsets of annotated data? *ii*) How to validate that the label consistency was recovered by the correction?

Another example is SCIERC (Luan et al., 2018) (cited ~ 50 times) which is a multi-task (including NER) benchmark in AI domain. It has 1,861 sentences for training, 455 for dev, and 551 for test. When we looked at the false predictions given by SCIEE which was a multi-task model released along with the SCIERC dataset, we found that as many as 147 (26.7% of the test set) sentences were not properly annotated. (We also recruited five annotators and counted a mistake when all the annotators report it.) Three examples are given in Table 1: two of them have wrong entity types; the third has a wrong span boundary. As shown in the experiments section, after the correction, the NER performance becomes more accurate and stable.

Table 1: Three examples to compare original and corrected annotation in the test set of the SCIERC dataset. If the annotation on the test set consistently followed the “codebook” that was used to annotate training data, the entities in the first two examples would be labelled as “Task” (not “Method”) for sure.

Original Examples	Corrected Examples
Starting from a DP-based solution to the [traveling salesman problem]Method, we present a novel technique ...	Starting from a DP-based solution to the [traveling salesman problem]Task, we present a novel technique ...
FERRET utilizes a novel approach to [Q/A]Method known as predictive questioning which attempts to identify ...	FERRET utilizes a novel approach to [Q/A]Task known as predictive questioning which attempts to identify ...
The goal of this work is the enrichment of [human-machine interactions]Task in a natural language environment.	The goal of this work is the [enrichment of human-machine interactions]Task in a natural language environment.

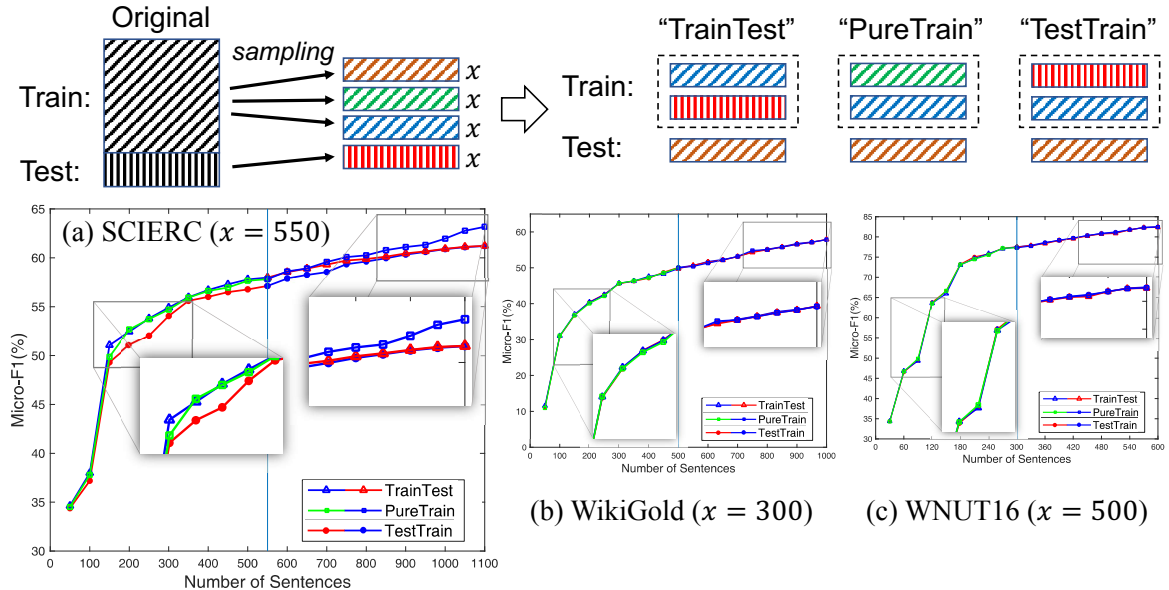


Figure 1: *Identifying label inconsistency of test set with training set*: We sample *three* exclusive subsets (of size x) from the training set (orange, green, and blue). We use one subset as the *new* test set (orange). We apply the SCIEE NER model on the new test set. We build three *new* training sets: *i*) “TrainTest” (blue-red), *ii*) “PureTrain” (green-blue), *iii*) “TestTrain” (red-blue). Results on SCIERC show that the test set (red) is *less* predictive of training samples (orange) than the training set itself (blue or green). This was not observed on two other datasets.

Besides the significant correction on the SCIERC dataset, our contributions in this work are as follows: *i*) an empirical, visual method to identify the label inconsistency between subsets of annotated data (see Figure 1), *ii*) a method to validate the label consistency of corrected data annotation (see Figure 2). Experiments show that they are effective on the CoNLL03 and SCIERC datasets.

2 Proposed Methods

2.1 A method to identify label inconsistency

Suppose the labeling processes on two parts of annotated data were consistent. They are likely to be equivalently predictive of each other. In other words, if we train a model with a set of samples from either part A or part B to predict a different set from part A , the performance should be similar.

Take SCIERC as an example. We were wondering whether the labels in the test set were consistent with those in the training set. Our method to identify the inconsistency is presented in Figure 1.

We sample *three* exclusive subsets (of size x) from the training set. We set $x = 550$ according to the size of the original test set. We use one of the subsets as the *new* test set. Then we train the SCIEE NER model (Luan et al., 2018) to perform on the new test set. We build three *new* training sets to feed into the model:

- “TrainTest”: first fed with one training subset and then the original test set;
- “PureTrain”: fed with two training subsets;
- “TestTrain”: first fed with the original test set and then one of the training subsets.

Results show that “TestTrain” performed the worst at the early stage because the quality of the

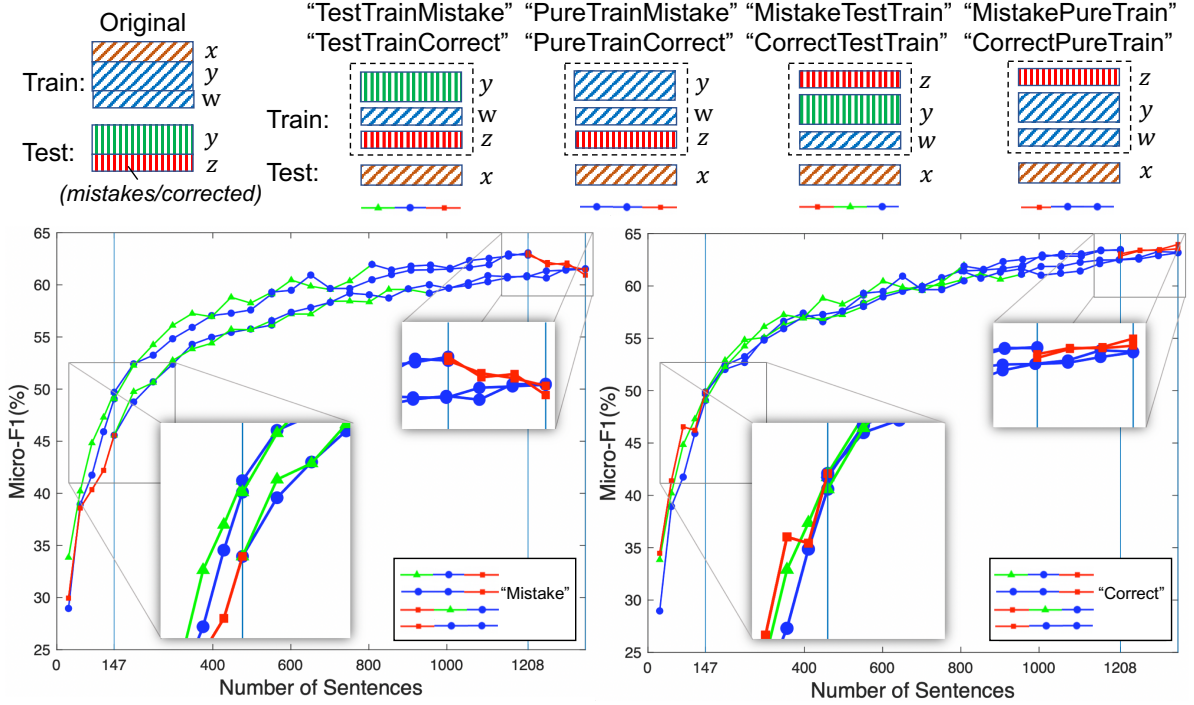


Figure 2: *Validating label consistency in corrected test set*: We corrected z of $y + z$ sentences in the test set. We sampled *three* exclusive subsets of size x , y , and w from the training set. We use the first subset (of size x) as the *new* test set. We build four *new* training sets as shown in the figure and feed them into the SCIIE model (at the top of the figure). Results show that the label mistakes (red parts of the curves on the left) do hurt the performance no matter fed at the beginning or later; and the corrected test set performs as well as the training set (on the right).

original test set is not reliable. In “TrainTest” the performance no longer improved when the model started being fed with the original test set. “PureTrain” performed the best. All the observations conclude that the original test set is less predictive of training samples than the training set itself. It may be due to the issue of label inconsistency. Moreover, we do not have such observations on two other datasets, WikiGold and WNUT16.

2.2 A method to validate label consistency after correction

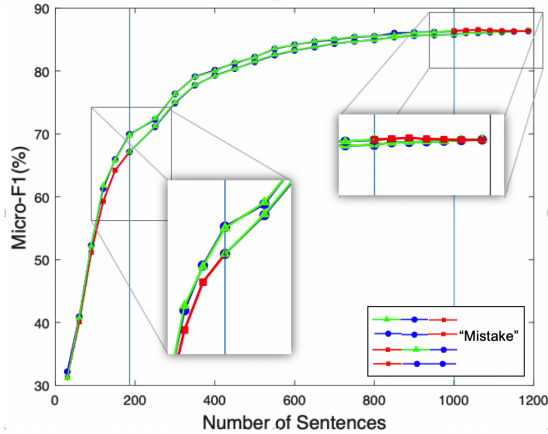
After we corrected the label mistakes, how could we empirically validate the recovery of label consistency? Again, we use a subset of training data as the new test set. We evaluate the predictability of the original wrong test subset, the corrected test subset, and the rest of the training set. We expect to see that the wrong test subset delivers weaker performance and the other two sets make comparable good predictions. Figure 2 illustrates this idea.

Take SCIIE as an example. Suppose we corrected z of $y + z$ sentences in the test set. The original wrong test subset (“Mistake”) and the corrected test subset (“Correct”) are both of size z .

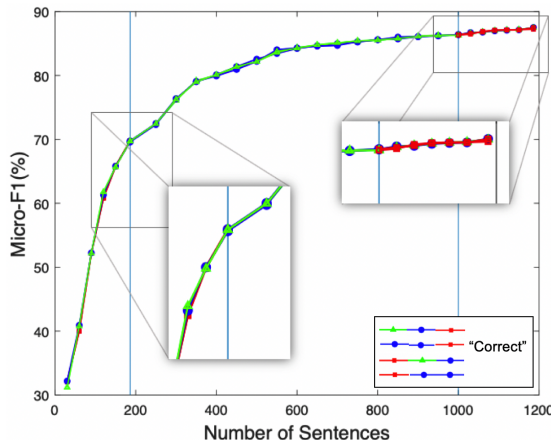
Here $z = 147$ and the original good test subset $y = 404$ (“Test”). We sampled *three* exclusive subsets of size x , y , and $w = 804$ from the training set (“Train”). We use the first subset (of size x) as the *new* test set. We build four *new* training sets and feed into the SCIIE model. Each new training set has $y + w + z = 1,355$ sentences.

- “TestTrainMistake”/“TestTrainCorrect”: the original good test subset, the third sampled training subset, and the original wrong test subset (or the corrected test subset);
- “PureTrainMistake”/“PureTrainCorrect”: the second and third sampled training subsets and the original wrong test subset (or the corrected test subset);
- “MistakeTestTrain”/“CorrectTestTrain”: the original wrong test subset (or the corrected test subset), the original good test subset, and the third sampled training subset;
- “MistakePureTrain”/“CorrectPureTrain”: the original wrong test subset (or the corrected test subset) and the second and third sampled training subsets.

Results show that the label mistakes (i.e., original wrong test subset) hurt the model performance



(a) Original with label mistakes



(b) Corrected

Figure 3: Identifying label inconsistency and validating the consistency in the original & corrected CoNLL03.

whenever being fed at the beginning or later. The corrected test subset delivers comparable performance with the original good test subset and the training set. This demonstrates the label consistency of the corrected test set with the training set.

3 Experiments

3.1 Results on SCIERC

The visual results of the proposed methods have been presented in Section 2. Here we deploy five state-of-the-art NER models to investigate their performance on the corrected SCIERC dataset. The NER models are BiLSTM-CRF (Lample et al., 2016), LM-BiLSTM-CRF (Liu et al., 2018), single-task and multi-task SCIIE (Luan et al., 2018), and multi-task DyGIE (Luan et al., 2019).

As shown in Table 2, all NER models deliver better performance on the corrected SCIERC than the original dataset. So the training set is more consistent with the fixed test set than the original wrong test set. In future work, we will explore

Table 2: Five NER models perform consistently better on the corrected SCIERC than on the original dataset.

Method	Corrected SCIERC			Original SCIERC		
	P	R	F1	P	R	F1
BiLSTM-CRF	58.35	47.95	52.64	56.13	48.07	51.79
LM-BiLSTM-CRF	62.78	58.20	60.40	59.15	57.15	<u>58.13</u>
SCIIE-single	71.20	62.88	66.79	65.77	60.90	<u>63.24</u>
SCIIE-multi	72.66	63.22	67.61	67.66	61.72	<u>64.56</u>
DyGIE-multi	69.64	67.02	68.31	65.09	65.28	<u>65.18</u>

more baselines in the leaderboard.

3.2 Results on CoNLL03

Based on the correction contributed by (Wang et al., 2019), we use the proposed method to justify label inconsistency though the label mistakes take “only” 5.38%. It also validates the label consistency after recovery. Figure 3(a) shows that starting with the wrong labels in the original test set makes the performance worse than starting with the training set or the good test subset. After label correction, this issue is fixed in Figure 3(b).

4 Related Work

NER is typically cast as a sequence labeling problem and solved by models integrate LSTMs, CRF, and language models (Lample et al., 2016; Liu et al., 2018; Zeng et al., 2019, 2020). Another idea is to generate span candidates and predict their type. Span-based models have been proposed with multi-task learning strategies (Luan et al., 2018, 2019). The multiple tasks include concept recognition, relation extraction, and co-reference resolution.

Researchers notice label mistakes in many NLP tasks (Manning, 2011; Wang et al., 2019; Eskin, 2000; Květoň and Oliva, 2002). For instance, it is reported that the bottleneck of the POS tagging task is the consistency of the annotation result (Manning, 2011). People tried to detect label mistakes automatically and minimize the influence of noise in training. The mistake re-weighting mechanism is effective in the NER task (Wang et al., 2019). We focus on visually evaluating the label consistency.

5 Conclusion

We presented an empirical method to explore the relationship between label consistency and NER model performance. It identified the label inconsistency of test data in SCIERC and CoNLL03 datasets (with 26.7% and 5.4% label mistakes). It validated the label consistency in

multiple sets of NER data annotation on two benchmarks, CoNLL03 and SCIERC.

Acknowledgements

This work is supported by National Science Foundation IIS-1849816 and CCF-1901059.

References

- Eleazar Eskin. 2000. Detecting errors within a corpus using anomaly detection. In *Proceedings of the 1st North American chapter of the Association for Computational Linguistics conference*, pages 148–153. Association for Computational Linguistics.
- Tianwen Jiang, Qingkai Zeng, Tong Zhao, Bing Qin, Ting Liu, Nitesh V Chawla, and Meng Jiang. 2020. Biomedical knowledge graphs construction from conditional statements. *IEEE/ACM transactions on computational biology and bioinformatics*, 18(3):823–835.
- Pavel Květoň and Karel Oliva. 2002. (semi-)automatic detection of errors in PoS-tagged corpora. In *COLING 2002: The 19th International Conference on Computational Linguistics*.
- Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. *arXiv preprint arXiv:1603.01360*.
- Liyuan Liu, Jingbo Shang, Xiang Ren, Frank Fangzheng Xu, Huan Gui, Jian Peng, and Jiawei Han. 2018. Empower sequence labeling with task-aware neural language model. In *Thirty-Second AAAI Conference on Artificial Intelligence*.
- Yi Luan, Luheng He, Mari Ostendorf, and Hannaneh Hajishirzi. 2018. Multi-task identification of entities, relations, and coreference for scientific knowledge graph construction. *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.
- Yi Luan, Dave Wadden, Luheng He, Amy Shah, Mari Ostendorf, and Hannaneh Hajishirzi. 2019. A general framework for information extraction using dynamic span graphs. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.
- Christopher D Manning. 2011. Part-of-speech tagging from 97% to 100%: is it time for some linguistics? In *International conference on intelligent text processing and computational linguistics*.
- Erik F Sang and Fien De Meulder. 2003. Introduction to the conll-2003 shared task: Language-independent named entity recognition. *arXiv preprint cs/0306050*.
- 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 (EMNLP-IJCNLP)*.
- Wenhao Yu, Mengxia Yu, Tong Zhao, and Meng Jiang. 2020. Identifying referential intention with heterogeneous contexts. In *Proceedings of The Web Conference 2020*.
- Qingkai Zeng, Jinfeng Lin, Wenhao Yu, Jane Cleland-Huang, and Meng Jiang. 2021. Enhancing taxonomy completion with concept generation via fusing relational representations. In *ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD)*.
- Qingkai Zeng, Mengxia Yu, Wenhao Yu, Jinjun Xiong, Yiyu Shi, and Meng Jiang. 2019. Faceted hierarchy: A new graph type to organize scientific concepts and a construction method. In *Proceedings of the Thirteenth Workshop on Graph-Based Methods for Natural Language Processing (TextGraphs-13)*.
- Qingkai Zeng, Wenhao Yu, Mengxia Yu, Tianwen Jiang, Tim Wenginger, and Meng Jiang. 2020. Tri-train: Automatic pre-fine tuning between pre-training and fine-tuning for sciner. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*.