Improving Named Entity Recognition in Telephone Conversations via Effective Active Learning with Human in the Loop

Md Tahmid Rahman Laskar, Cheng Chen, Xue-Yong Fu, Shashi Bhushan TN
Dialpad Canada Inc.
Vancouver, BC, Canada
{tahmid.rahman,cchen,xue-yong,sbhushan}@dialpad.com

Abstract

Telephone transcription data can be very noisy due to speech recognition errors, disfluencies, etc. Not only that annotating such data is very challenging for the annotators, but also such data may have lots of annotation errors even after the annotation job is completed, resulting in a very poor model performance. In this paper, we present an active learning framework that leverages human in the loop learning to identify data samples from the annotated dataset for re-annotation that are more likely to contain annotation errors. In this way, we largely reduce the need for data re-annotation for the whole dataset. We conduct extensive experiments with our proposed approach for Named Entity Recognition and observe that by re-annotating only about 6% training instances out of the whole dataset, the F1 score for a certain entity type can be significantly improved by about 25%.

1 Introduction

We describe a Named Entity Recognition (NER) system that needs to provide realtime functionality in a commercial communication-as-a-service (CaaS) platform such as displaying information related to the named entities to a customer support agent during a call with a customer. The primary focus for this NER system is to identify *product* and *organization* type entities that appear in English business telephone conversation transcripts. Since these transcripts are produced by an automatic speech recognition (ASR) system, they are inherently noisy due to the nature of spoken communication as well as limitations of the ASR system, resulting in dysfluencies, filled pauses, and lack of information related to punctuation and case (Fu et al., 2021). These issues make the annotation of such noisy datasets very challenging for the annotators, making the annotation job for such data more difficult than datasets that contain typed text (Meng et al., 2021; Malmasi et al., 2022).

To our best knowledge, the publicly available NER datasets mostly contain typed text. Moreover, there are no existing NER datasets that match the characteristics of the ASR transcripts in the domain of business telephone conversations (Li et al., 2020). Thus, training an NER model in the context of business telephone transcripts requires a large annotated version of such data (Fu et al., 2022b) consisting of *product* or *organization* type entites that usually appear in business conversations (Meng et al., 2021). However, the difficulties to understand noisy data may lead to annotation errors even in the human annotated version of such data.

To address the above issues, in this paper, we present an active learning (Ren et al., 2021) framework for NER that effectively sample instances from the annotated training data that are more likely to contain annotation errors. Moreover, we also address a very challenging task that is not widely studied in most of the existing NER datasets, distinguishing between *organization* and *product* type entities. In addition, since the NER model needs to provide realtime functionality in a commercial CaaS product, we show how data re-annotation through our active learning framework can even help smaller models to obtain impressive performance.

2 Related Work

In recent years, transformer-based pre-trained language models have significantly improved the performance of NER across publicly available academic datasets leading to a new state-of-the-art performance (Devlin et al., 2019; Yamada et al., 2020; Meng et al., 2021). However, there remain several issues in these existing benchmark datasets. For instance, most of these datasets are constructed from articles or the news domain, making these datasets quite well-formed with punctuation and casing information, along with having rich context around

Proceedings of the Fourth Workshop on Data Science with Human-in-the-Loop (Language Advances), pages 88 - 93
December 8, 2022 ©2022 Association for Computational Linguistics

Sample Utterance
Exactly, yes, absolutely. Health Insurance USA will pay for your physiotherapy and the prescription that you were taking previously um-hum only thing you need to do is to go to our branchand fill out the form required by the insurance company and let me know if you have any questions about the claim process.
You gotta send email to the Netflix team and ask for refund.
The contending discussion of the this guy would setting the TP and other services into the university of Toronto. The one where we just got to the we're just about to serve the meeting vegetables and last week's conference and then, zoom crash.

Table 1: Example Utterances in Noisy Business Conversations.

the entities. Meanwhile, it has been observed recently that the NER models tend to memorize the entities in the training data, resulting in improved entity recognition when those entities also appear in the test data (Lin et al., 2021). Furthermore, it has been found that models trained on such academic datasets tend to perform significantly worse on unseen entities as well as on noisy text (Bodapati et al., 2019; Bernier-Colborne and Langlais, 2020; Malmasi et al., 2022).

To investigate the above issues, Lin et al. (2021) created adversarial examples via replacing target entities with other entities of the same semantic class in Wikidata and observed that existing state-of-the-art models mostly memorized in-domain entity patterns instead of reasoning from the context. Since the dataset that we study in this paper is constructed from real-world business phone conversations, there are many entities that may appear only in our test set as well as in real-world production settings that do not appear in the training set.

Note that due to the presence of speech recognition errors as well as annotation errors, training a model to be more generalized to detect the unseen entities in noisy conversations is fundamentally more challenging than above body of work. In addition, annotating such noisy datasets are also more difficult and expensive (Fu et al., 2021). In such scenarios, techniques such as Active Learning (Ren et al., 2021), that samples only a few instances from the given dataset for annotation could be very effective to train deep learning models. In this paper, we also investigate how active learning can be leveraged to fix the data annotation errors via utilizing an effective human in the loop learning.

3 Dataset Construction

The dataset used in this paper is constructed from transcripts produced by an ASR system (see Table 1 for some sample utterances). Thus, our dataset may miss many punctuation marks while only consisting of partial casing information. This makes the entity recognition on this dataset very challenging since casing information gives a very strong hint of a token being a named entity (Bodapati et al., 2019; Mayhew et al., 2019).

# Examples	Train	Dev	Test
Utterances	55,522	7,947	15,814
Person tags	34,859	4,825	10,270
Product tags	36,553	5,292	10,851
Organization tags	23,942	3,720	6,785
GPE Location tags	22,697	3,309	6,533

Table 2: Labeled in-domain dataset class distribution.

For data annotation, at first, we sampled 78,983 utterances containing human to human business telephone conversation transcripts and sent to Appen[1] for annotation. We asked the annotators in Appen to label four types of entities: *person name*, *product*, *organization*, and *geopolitical location*. The detailed statistic of the dataset labeled by Appen is shown in Table 2.

The initial selection criteria for the annotators is that they were required to be fluent in English. Moreover, the annotators had to pass a screening test where they were given some sample utterances to annotate the named entities. Based on their performance in the screening test, they were selected for the annotation job to ensure better quality for data annotation.

4 Proposed Active Learning Framework

Suppose, there is a sequence $S = s_1, s_2, ..., s_n$ containing n words. For each token s_i, the sequence tagging model will assign the most relevant tag t_j (based on the highest probability score predicted by the model) from a list of m tags $T = t_1, t_2, t_3, ..., t_m$. While constructing a sequence tagging dataset (i.e., NER dataset) from our business conversation data, we ask the annotators to annotate each token with the most relevant tag. Due to the nature of our dataset, there is a high risk of annotation errors. Thus, in this paper,

[1] https://appen.com/

Algorithm 1 The Active Learning Framework

```
$Folds$: {1, 2, 3, 4, 5}
$ReAnnotationSet$:{}
$T$: $Threshold\_Value$

for $Fold$ in $Folds$ do
    $PredictionSet \leftarrow samplePredictionData()$
    $TrainingSet \leftarrow sampleTrainingData()$
    $Model \leftarrow trainModel(TrainingSet)$
    for $utterance$ in $PredictionSet$ do
        $results \leftarrow Model.Predict(utterance)$
        for $entity, p\_tag$ in $results$ do
            $g\_tag \leftarrow getGoldTag(entity, utterance)$
            $p\_prob\_score \leftarrow getProbScore(p\_tag)$
            $g\_prob\_score \leftarrow getProbScore(g\_tag)$
            if $p\_prob\_score - g\_prob\_score > T$ then
                $ReAnnotationSet.add(utterance)$
            end if
        end for
    end for
end for
```

our objective is to reduce the annotation errors that may occur in noisy datasets via utilizing active labeling. Below, we demonstrate our proposed active learning framework.

N-fold Experiments: Given a dataset containing N examples, M fold experiments can be run in the following way to select some samples from the training set that are more likely to contain annotation errors. In each fold, use X% data from the training set as the prediction data (without replacement). Also, the prediction set in each fold should only contain distinct examples (i.e., the examples that do not appear in any other fold's prediction set) such that combining all the prediction sets together covers all the training instances.

In Algorithm 1, we present our framework via demonstrating a 5 fold experiments. The sampled data in each fold will be as follows: 80% data in each fold will contain the training set while 20% data in each fold will contain the prediction set. Moreover, the prediction data in each fold should contain those examples only that do not appear in any other prediction set.

For model training, we fine-tune a BERT-based-cased model on each fold of the training data. In this way, we fine-tune 5 BERT-based-cased models on the training data of 5 different folds. Then each trained model is utilized to predict the NER tags (p_tag refers the predicted tag in Algorithm 1) on the prediction data in their respective folds. Finally, we select some instances from the prediction set for re-annotation based on the following method.

Probability Thresholding: We utilize predicted probabilities to select instances from the prediction set for re-annotation. For an $entity$ in an example utterance in the predicted set of the training data, if the predicted tag is p_tag, with the probability score predicted by the NER model for that tag is p_prob_score and the predicted probability score for the gold entity is g_prob_score, then if the probability score difference between p_prob_score and g_prob_score is more than a threshold T (where p_prob_score > g_prob_score), then we add that utterance in our data re-annotation set.

Human in the Loop: Once the data re-annotation set is constructed, it can be sent to the annotators for re-labeling. At first, the annotators can decide whether the given tags for an utterance are correct or not. If they think it is incorrect, then they are asked to re-annotate the utterance. In this way, we speed up the annotation process. Note that for data re-annotation, Labelbox[2] was used.

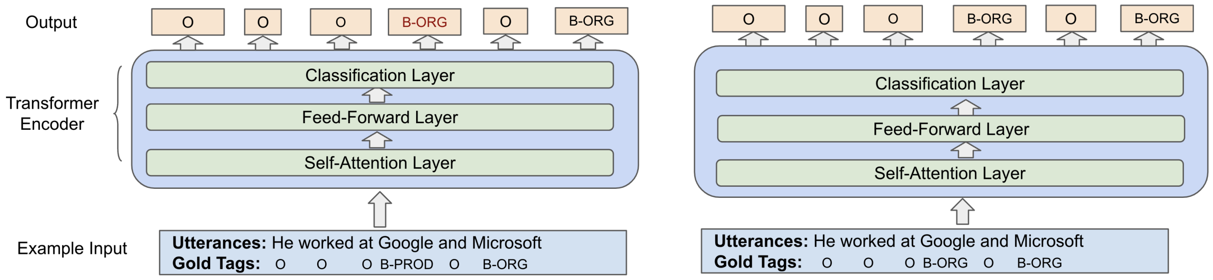
5 Model Architecture

We use two models in two stages to run our experiments. In stage 1, we fine-tune a BERT-base-cased model (Devlin et al., 2019; Laskar et al., 2019) on each fold of the dataset to identify the instances that are more likely to contain annotation errors. After re-annotating those instances, we run another experiment (i.e., stage 2) in the updated version of the training set containing the instances that are selected for re-annotation along with other instances (i.e., the instances that were not selected for re-annotation). Note that the model trained on stage 2 is the one that is used for production deployment. For this reason, we choose the DistilBERT-base-cased (Sanh et al., 2019) model for stage 2 since it is more efficient than BERT while also being significantly smaller, making it more applicable for industrial scenarios. A general overview of our proposed approach is shown on Figure 1.

5.1 Stage 1: N-fold BERT Fine-Tuning for Re-Annotation

In this section, we describe our N-fold experiments with the BERT-base-cased model to sample the instances for data re-annotation. Though our proposed Active Learning Framework is applicable for all type of entities: *Product*, *Organization*, *GPE Location*, and *Person*; in practice, we observe during

[2] https://labelbox.com/



(a) Stage 1: N-Fold Fine-Tuning of BERT on the originally annotated data for Re-Annotation

(b) Stage 2: DistilBERT Fine-Tuning on the updated version of the dataset after re-annotation for Deployment

Figure 1: Our proposed approach: (a) at first, we do fine-tuning using the BERT on the originally annotated dataset to identify utterances where the difference between the predicted probability score for the Organization tag and the gold tag is above a certain threshold T, (b) Next, we fine-tune the DistilBERT model on the updated version of the dataset that is re-annotated in Stage 1. For the given utterance "He worked at Google and Microsoft", suppose the original tag for the token "Google" was B-PROD but during N-fold experiments we get the predicted tag as B-ORG, if the predicted probability score difference between the predicted tag and gold tag is above threshold T, then we select that utterance for re-annotation.

Type	**Original Dataset**			**Re-Annotated Dataset**		
	Precision	**Recall**	**F1**	**Precision**	**Recall**	**F1**
Person	79.30	**79.68**	**79.73**	**80.82**	78.20	79.49
GPE Location	79.78	79.86	79.82	**82.72**	**79.89**	**81.28**
Product	75.94	74.08	75.01	**77.26**	**74.82**	**76.02**
Organization	48.27	42.45	45.18	**60.47**	**52.50**	**56.21**
Overall (all 4 types)	83.49	81.47	82.41	**85.10**	**82.26**	**83.66**

Table 3: Performance of DistilBERT in the original version and the re-annotated version of the dataset.

our N-fold experiments that most of the times when there is a huge difference between the probability score of the predicted tag and the gold tag is when the predicted tag is *organization* type tag. Thus, when sampling the data for re-annotation, we focus on those utterances that are more likely to contain annotation errors for *organization* type entities and ask the annotators to re-annotate them. Moreover, focusing on annotation errors in *organization* type entities also helps our model to better distinguish between *product* and *organization* type entities.

In this way, we sample 3166 utterances for re-annotation out of 55,222 training samples that are more likely to contain annotation errors for *organization* type entities. To sample these utterances, we define a threshold and measure the difference in probability score between the predicted *'organization' tag* and the *gold tag*. If the probability score difference is above that threshold, we consider this utterance as more likely to contain annotation errors and select it for re-annotation. For this experiment, we set the threshold[3] value $T = 2.0$.

5.2 Stage 2: DistilBERT Fine-Tuning for Production Deployment

Since our goal is to deploy an NER model in production for real-time inference while utilizing limited computational resources, we need to choose a model that is fast enough and also requires minimum computational memory. For this reason, we choose DistilBERT (Sanh et al., 2019) as it is much faster and smaller than the original BERT (Devlin et al., 2019) model (though a bit less accurate).

After re-annotation is done for the sampled utterances in Stage 1, we update the labels of those utterances. Then, we fine-tune the DistilBERT-base-cased model in the re-annotated training data.

6 Results and Analyses

We conduct experiments in the original version of the dataset as well as the re-annotated version of the dataset using DistilBERT. During experiments, we run 5 $epochs$ with the $training\_batch\_size$ set to 64. The $learning\_rate$ was set to $1e - 4$ with the $max\_sequence\_length$ being set to 200.

We show our experimental results in Table 3 to find that for all entity types, the Precision score is increased when the model is trained on the re-

[3] We also tried other values but $T = 2$ performed the best.

Type	DistilBERT			DistilBERT$_{\text{dtft}}$		
	Precision	Recall	F1	Precision	Recall	F1
Person	80.82	78.20	79.49	**82.06**	**82.46**	**82.26**
GPE Location	82.72	79.89	81.28	**82.90**	**83.29**	**83.09**
Product	77.26	74.82	76.02	77.23	**75.44**	**76.33**
Organization	60.47	52.50	56.21	**61.97**	**55.55**	**58.59**
Overall (all 4 types)	85.10	82.26	83.66	**85.41**	**84.04**	**84.73**

Table 4: Performance of DistilBERT and DistilBERT$_{\text{dtft}}$ in the re-annotated version of the dataset.

annotated version of the dataset. Moreover, except the Person type entity, Recall and F1 scores are also improved. Out of all 4 types of entities, we observe the highest performance improvement in terms of the *organization* type entity. This is expected since we sample data for re-annotation targeting the annotation errors in *organization* type entities. Meanwhile, we observe improvement in most other entities in addition to *organization* since the annotators were also asked to re-annotate the utterance even if there are annotation errors on any other entity types. By only re-annotating about 6% of the training data using our proposed active learning framework, we observe improvement: i) for *Organization*: 25.27%, 23.67, 24.41%, and (ii) *Overall (for all 4 types)*: 0.73%, 0.97%, and 1.52%, in terms of Precision, Recall, and F1 respectively.

The performance gain using our proposed active learning framework also makes this technique applicable for production deployment. To further improve the performance, we utilize the *distill-then-fine-tune (dtft)* architecture from Fu et al. (2022b) that achieves impressive performance on noisy data. Similar to their knowledge distillation (Hinton et al., 2015; Fu et al., 2022b) technique, we first fine-tune the teacher LUKE (Yamada et al., 2020) model on our re-annotated training set and generate pseudo labels for 483,766 unlabeled utterances collected from telephone conversation transcripts. Then, we fine-tune the student DistilBERT model in this large dataset of pseudo labels as well as the re-annotated training set via leveraging the two-stage fine-tuning mechanism (Fu et al., 2022b; Laskar et al., 2022c). We show the result in Table 4 to find that the DistilBERT$_{\text{dtft}}$ model further improves the performance and so we deploy this model in production.

7 Conclusion

In this paper, we propose an active learning framework that is very effective to fix the annotation errors in a noisy business conversation data. By sampling only about 6% of the training data for re-annotation, we observe a huge performance gain in terms of Precision, Recall, and F1. Moreover, re-annoating the data using the proposed technique also helps the NER model to better distinguish between *product* and *organization* type entities in noisy business conversational data. These findings further validate that our proposed approach is very effective in limited budget scenarios to alleviate the need of human re-labeling of a large amount of noisy data. We also show that a smaller-sized DistilBERT model can be effectively trained on such data and deployed in a minimum computational resource environment. In the future, we will investigate the performance of our proposed technique on other entity types, as well as on other tasks (Fu et al., 2022a; Laskar et al., 2022a,b) similar to NER (Fu et al., 2022b) containing noisy data.

Ethics Statement

The data used in this research is comprised of machine generated utterances. To protect user privacy, sensitive data such as personally identifiable information (e.g., credit card number, phone number) were removed while collecting the data. We also ensure that all the annotators are paid with adequate compensation. There is a data retention policy available for all users so that data will not be collected if the user is not consent to data collection. Since our model is doing classification to predict the named entities in telephone transcripts, incorrect predictions will not cause any harm to the user besides an unsatisfactory experience. While annotator demographics are unknown and therefore may introduce potential bias in the labelled dataset, the annotators are required to pass a screening test before completing any labels used for experiments, thereby mitigating this unknown to some extent.

ACKNOWLEDGEMENTS

We would like to thank Anne Paling and Kevin Sanders for their help with the data annotation job. We also thank the anonymous reviewers for their excellent review comments.

References

Gabriel Bernier-Colborne and Philippe Langlais. 2020. Hardeval: Focusing on challenging tokens to assess robustness of ner. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 1704–1711.

Sravan Bodapati, Hyokun Yun, and Yaser Al-Onaizan. 2019. Robustness to capitalization errors in named entity recognition. In *Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019)*, pages 237–242.

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 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Xue-Yong Fu, Cheng Chen, Md Tahmid Rahman Laskar, Shashi Bhushan, and Simon Corston-Oliver. 2021. Improving punctuation restoration for speech transcripts via external data. In *Proceedings of the Seventh Workshop on Noisy User-generated Text (W-NUT 2021)*, pages 168–174.

Xue-Yong Fu, Cheng Chen, Md Tahmid Rahman Laskar, Shayna Gardiner, Pooja Hiranandani, and Shashi Bhushan TN. 2022a. Entity-level sentiment analysis in contact center telephone conversations. *arXiv preprint arXiv:2210.13401*.

Xue-Yong Fu, Cheng Chen, Md Tahmid Rahman Laskar, Shashi Bhushan Tn, and Simon Corston-Oliver. 2022b. An effective, performant named entity recognition system for noisy business telephone conversation transcripts. In *Proceedings of the Eighth Workshop on Noisy User-generated Text (W-NUT 2022)*, pages 96–100.

Geoffrey Hinton, Oriol Vinyals, Jeff Dean, et al. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2(7).

Md Tahmid Rahman Laskar, Cheng Chen, Jonathan Johnston, Xue-Yong Fu, Shashi Bhushan TN, and Simon Corston-Oliver. 2022a. An auto encoder-based dimensionality reduction technique for efficient entity linking in business phone conversations. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 3363–3367.

Md Tahmid Rahman Laskar, Cheng Chen, Aliaksandr Martsinovich, Jonathan Johnston, Xue-Yong Fu, Shashi Bhushan Tn, and Simon Corston-Oliver. 2022b. BLINK with Elasticsearch for efficient entity linking in business conversations. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Industry Track*, pages 344–352. Association for Computational Linguistics.

Md Tahmid Rahman Laskar, Enamul Hoque, and Jimmy Xiangji Huang. 2022c. Domain adaptation with pre-trained transformers for query-focused abstractive text summarization. *Computational Linguistics*, 48(2):279–320.

Md Tahmid Rahman Laskar, Enamul Hoque, and Jimmy Xiangji Huang. 2019. Utilizing bidirectional encoder representations from transformers for answer selection. In *International Conference on Applied Mathematics, Modeling and Computational Science*, pages 693–703. Springer.

Jing Li, Aixin Sun, Jianglei Han, and Chenliang Li. 2020. A survey on deep learning for named entity recognition. *IEEE Transactions on Knowledge and Data Engineering*, 34(1):50–70.

Bill Yuchen Lin, Wenyang Gao, Jun Yan, Ryan Moreno, and Xiang Ren. 2021. Rockner: A simple method to create adversarial examples for evaluating the robustness of named entity recognition models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3728–3737.

Shervin Malmasi, Anjie Fang, Besnik Fetahu, Sudipta Kar, and Oleg Rokhlenko. 2022. Multiconer: a large-scale multilingual dataset for complex named entity recognition. *arXiv preprint arXiv:2208.14536*.

Stephen Mayhew, Tatiana Tsygankova, and Dan Roth. 2019. ner and pos when nothing is capitalized. 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 2019, Hong Kong, China, November 3-7, 2019*, pages 6255–6260. Association for Computational Linguistics.

Tao Meng, Anjie Fang, Oleg Rokhlenko, and Shervin Malmasi. 2021. Gemnet: Effective gated gazetteer representations for recognizing complex entities in low-context input. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1499–1512.

Pengzhen Ren, Yun Xiao, Xiaojun Chang, Po-Yao Huang, Zhihui Li, Brij B Gupta, Xiaojiang Chen, and Xin Wang. 2021. A survey of deep active learning. *ACM computing surveys (CSUR)*, 54(9):1–40.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of BERT: smaller, faster, cheaper and lighter. *CoRR*, abs/1910.01108.

Ikuya Yamada, Akari Asai, Hiroyuki Shindo, Hideaki Takeda, and Yuji Matsumoto. 2020. LUKE: deep contextualized entity representations with entity-aware self-attention. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 6442–6454. Association for Computational Linguistics.