An Effective, Performant Named Entity Recognition System for Noisy Business Telephone Conversation Transcripts

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

Abstract

We present a simple yet effective method to train a named entity recognition (NER) model that operates on business telephone conversation transcripts that contain noise due to the nature of spoken conversation and artifacts of automatic speech recognition. We first fine-tune LUKE, a state-of-the-art Named Entity Recognition (NER) model, on a limited amount of transcripts, then use it as the teacher model to teach a smaller DistilBERT-based student model using a large amount of weakly labeled data and a small amount of human-annotated data. The model achieves high accuracy while also satisfying the practical constraints for inclusion in a commercial telephony product: realtime performance when deployed on cost-effective CPUs rather than GPUs.

1 Introduction

We describe a named entity recognition (NER) system that identifies entities mentioned in English business telephone conversations. The input to the NER system is transcripts produced by an automatic speech recognition (ASR) system. These transcripts are inherently noisy due to the nature of spoken communication and due to the limitations of the ASR system. The transcripts contain dysfluencies, false starts, filled pauses, they lack punctuation information and have incomplete information about case.

Because there was no pre-existing annotated data set publicly available that matched the characteristics of the ASR transcripts in the domain of business telephone conversations (Li et al., 2020), the NER model is required to be trained on a large dataset containing telephone conversations to effectively detect named entities in such noisy data. Moreover, the NER model needs to provide real-time functionality in a commercial communication-as-a-service (CaaS) product such as displaying information related to the named entities to a customer support agent during a call with a customer. The deployed system was therefore required to be fast (less than 200ms inference time) but economical (able to operate on CPU, rather than more expensive GPUs).

To address the above issues, in this paper, we present a simple yet effective method, *distill-then-fine-tune*, to transfer knowledge from a large and complex model to a small and simple model while reaching a similar performance as the large model. More specifically, we fine-tune a state-of-the-art NER model, LUKE (Yamada et al., 2020), on our limited amount of noisy telephone conversations and predict the labels of a large amount of unlabeled conversations, denoted as distillation data. The smaller model is then trained on the distillation data using pseudo-labels. We conduct extensive experiments with our proposed approach and observe that our distilled model achieves 75x inference speed boost while reserving 99.09% F1 score of its teacher. This makes our proposed approach very effective in limited budget scenarios as it does not require the annotation of a huge amount of noisy data that would otherwise be required to fine-tune simpler transformers on downstream tasks.

2 Related Work

NER is often framed as a sequence labeling problem (Huang et al., 2015; Akbik et al., 2018) where a model is used to predict the entity type of each token. Previously, various models based on the recurrent neural network architecture have been widely used for this task. In recent years, pre-trained language models have been employed to perform the NER task where a new prediction layer is added into the pre-trained model to fine-tune for sequence labeling (Devlin et al., 2019).

More recently, (Yamada et al., 2020) proposed a new approach to provide the contextualized representations of words and entities based on a bidirectional transformer. In their proposed model, LUKE, they treat words and entities in a given context

Proceedings of the 2022 COLING Workshop: The 8th Workshop on Noisy User-generated Text (W-NUT 2022), pages 96–100

Type	Utterances	Person	Prod/Org	Location
Train	16124	4852	4443	4135
Dev	2292	682	627	629
Test	4497	1382	1274	1151

Table 1: Labeled in-domain dataset class distribution. The numbers under each entity type represent number of utterances containing the specific type.

as independent tokens, and output the contextualized representations of them. The LUKE model achieved impressive performance in various entity-related tasks. However, this model is inherently slow due to its complex architecture and so it is not applicable for usage in production environments in a limited computational budget scenario.

In scenarios where the computational budget is limited, using a smaller model that can mimic the behaviour of the large model can be used. Knowledge distillation (Hinton et al., 2015) is one such technique where a large model is compressed into a small model. One prominent approach for Knowledge Distillation that has been used in recent years is the work of (Tang et al., 2019), where they proposed a task specific knowledge distillation method to show that using an additional unlabeled transfer dataset can augment the training set for more effective knowledge transfer. However, most prior work that leveraged such knowledge distillation techniques focused on typed input, whereas the amount of work that leveraged knowledge distillation for noisy texts (e.g., telephone conversation transcripts) is very limited (Gou et al., 2021). Motivated by the advantages of knowledge distillation, in this work, we also leverage knowledge distillation to address the computational issues that occur while utilizing large state-of-the-art language models in a limited computational environment, while minimizing the amount of noisy data that must be human-annotated for use during fine-tuning.

3 Datasets

In this section, we first introduce the in-domain training data (noisy human-to-human conversations) that we sampled and annotated to train the teacher model. Then, we describe the data used for knowledge distillation of the student model.

3.1 In-domain Data Annotation

Since our in-domain dataset is sampled from transcripts produced by an ASR system, the dataset does not contain any punctuation marks and only contains partial casing information. This makes the property of our dataset fundamentally different from the data that most pre-trained models are trained on. This also makes the task more difficult since upper-cased words are a very strong hint of a token being a named entity (Mayhew et al., 2019).

For data annotation, we sampled 26,000 utterances from telephone conversation transcripts and had them annotated by Appen[1]. Four types of named entities were labeled by the annotators: *person name*, *product or organization*, *geopolitical location*, and *none*. The detailed statistic of this dataset labeled by Appen is shown in Table 1.

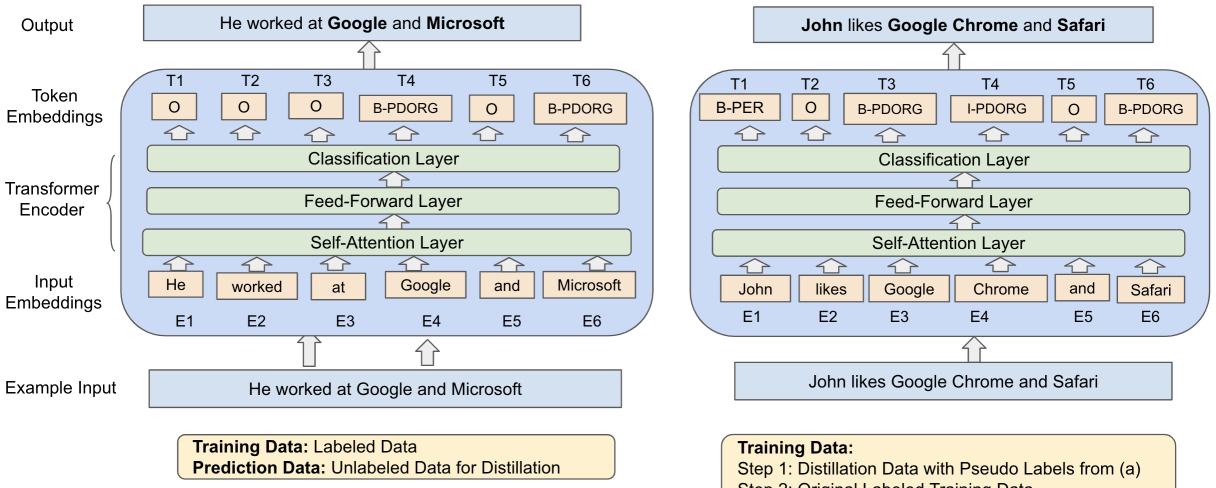
3.2 In-domain Distillation Data

Our goal is to reduce the amount of human annotated data in the training set. For this purpose, we perform knowledge distillation that transfers knowledge from a large and complex teacher model to a small and simple student model. Since the student model is expected to be much simpler than the teacher model, it requires a large amount of labeled training data. In addition, due to the sparsity of named entities, the model cannot learn too much from randomly sampled utterances where most of them may not contain any named entities. We address this issue by using the spaCy[2] NER model to select utterances that are highly likely to contain at least one named entity of a type we are interested in. Specifically, we only used four entity types relevant to this study from the spaCy model: PERSON, ORG, GPE, PRODUCT. This sampling method produced 483, 766 unlabeled utterances from business telephone conversation transcripts and largely increased the information density in the data. However, annotating this huge amount of unlabeled data would be a prohibitively costly process. To tackle this problem, we use the trained teacher model to predict the labels of these utterances. In this way, the teacher model provides the pseudo-labels of a large unlabeled noisy dataset to alleviate the need of human annotation for such data. We use this large noisy speech data with pseudo-labels as the distillation data to train the student model. The statistics of this dataset is listed in Table 2.

4 Our Proposed Approach

In this section, we first describe the architectures of the teacher and student models. We then de-

[1] https://appen.com/, accessed on January 4, 2022.
[2] https://spacy.io/api/entityrecognizer



(a) Stage 1: Fine Tuning Teacher on Labeled Data

(b) Stage 2: Two-Step Fine Tuning of Student on the Distillation Data (step 1) and the Training data (step 2)

Figure 1: Our knowledge distillation approach: (a) first, fine-tune the teacher model (LUKE) on the labeled dataset, and generate the pseudo-labels of a huge amount of unlabeled data for distillation. (b) Next, fine-tune the student model (DistilBERT) in two steps, **step 1:** on the distillation data having pseudo-labels that were generated in the previous step, and **step 2:** on the original labeled training data where the teacher model was also trained. Here, 'PDORG' denotes 'PROD/ORG', while 'Bold' font in the output layer denotes the entities tagged by the model.

Type	# Examples
Positive utterances	347,412
Negative utterances	136,354
Utterances containing *Person* tags	179,495
Utterances containing *Prod/Org* tags	97,857
Utterances containing *Location* tags	138,989

Table 2: Pseudo-labeled distillation data class distribution. "Positive utterances" are those that contain any of the 3 entity types, and "Negative utterances" are those that do not contain any of the 3 entity types. Here, '#' denotes 'Total number of'.

scribe our proposed knowledge distillation method, *distill-then-fine-tune*, that can be broken down into four steps: i) fine-tune the teacher model on the in-domain data, ii) sample distillation data from unlabeled examples, iii) perform distillation, and iv) fine-tune the student model. An overview of our proposed approach is illustrated in Figure 1.

Model Architecture: We use LUKE, a bidirectional transformer, that was pre-trained by (Yamada et al., 2020) on Wikipedia data to learn contextualized representations of words and entities. In LUKE, the input representation of a token (word or entity) is computed using three types of embedding: token embedding, position embedding, and entity type embedding. Token embedding, which is decomposed into two small matrices, represents the corresponding token. Position embedding represents the position of a token in a word sequence, while the entity type embedding represents whether the token is an entity. To further leverage the entity type embedding, an entity-aware self attention mechanism is used to handle interactions between entities in a given word sequence. Since LUKE is a large model that contains approximately 483M parameters (355M on its encoder and 128M for entity embeddings), we use it as the teacher to teach a student model.

For the student model, we adapt the DistilBERT (Sanh et al., 2019) model, a 6-layer bidirectional transformer encoder that was pre-trained for the language modeling task by Sanh et al. (2019). The DistilBERT model was initialized from its teacher BERT model by taking one layer out of two. It was pre-trained on the same corpus as BERT while using both the distillation loss and the masked language modelling loss. It contains approximately 66M parameters (approximately one seventh the size of the teacher model), making it more economical to deployment in a production environment with limited resources.

Distillation Method: Our goal is to build an NER system that can detect named entities in business conversations, but the LUKE model that we employ as a teacher model was pre-trained on written text, which is very different from noisy transcribed human-to-human conversations. To adapt

Model	F1 Score	Inference Time
LUKE_{ft}	86.07	2980ms
$\text{DistilBERT}_{\text{ft}}$	83.08	40ms
$\text{DistilBERT}_{\text{dtft}}$	85.29	40ms

Table 3: Performance of our proposed $\text{DistilBERT}_{\text{dtft}}$ models (fine-tuned on a large amount of distillation data and a small amount of in-domain human-annotated data) compared to the LUKE_{ft} and $\text{DistilBERT}_{\text{ft}}$ models that were fine-tuned only on the in-domain human-annotated data. Inference time is measured on a 2.20Ghz Intel Xeon CPU with sixteen virtual cores.

to the domain of business conversations, we first fine-tune the LUKE model on 16,124 in-domain human-annotated examples (see Section 3.1 for details). The resulting model is called LUKE_{ft}. The LUKE_{ft} model serves as the teacher that generates pseudo-labels for the distillation data (see Section 3.2 for details).

Next, we use a two-step fine-tuning approach for the student model (Fu et al., 2021; Laskar et al., 2022c). The student model is initialized with the pre-trained DistilBERT model. For step 1, we fine-tune the student model on the distillation data with pseudo-labels generated by the teacher. During the training stage, we use the cross entropy loss defined below.

$$L_{CE} = -\frac{1}{N}\sum_{n=1}^{N}\log\frac{e^{\hat{y}_{n,y_n}}}{\sum_{c=1}^{C}e^{\hat{y}_{n,c}}} \quad (1)$$

Here, N is the number of samples in a batch, and C denotes the number of classes. $\hat{y}_{n,c}$ is the logit of the c-th class in the n-th example, and $\hat{y}_{n,y_n}$ is the logit of the gold class in the n-th example.

For the final distillation step, we fine-tune the student model further on the in-domain human-annotated data. The resulting child model is termed **$\text{DistilBERT}_{\text{dtft}}$**.

5 Experiments

In this section, we describe our experimental settings and results.

5.1 Experimental Settings

Below, we discuss the baseline models and the training parameters used in our experiments.

Baselines: To compare the performance with our proposed model, we use the following baselines, ***(i) LUKE_{ft}***: The pre-trained LUKE model fine-tuned on our human-annotated in-domain training data, and ***(ii) DistilBERT_{ft}***: Similar to the other baseline, it was fine-tuned only on our human-annotated in-domain training data.

Training Parameters: For the teacher model, LUKE_{ft}, we set the batch size to 2, learning rate to 5×10^{-5}, and the number of epochs to 3. For the student DistilBERT model, we set the batch size to 32 and the learning rate to 5×10^{-5}, and the number of epochs to 5.

5.2 Results and Analyses

From Table 3, we see that the LUKE_{ft} model (fine-tuned on in-domain human-annotated data) achieves the highest F1 score, 86.07%, but with an inference time of 2980ms it is not practical for realtime applications.

The $\text{DistilBERT}_{\text{ft}}$ model (also fine-tuned only on the in-domain human-annotated data), with an inference time of 40ms is suitable for realtime application, but loses almost three percentage points of accuracy, reducing to an F1 score of 83.08%.

Our proposed **$\text{DistilBERT}_{\text{dtft}}$** model, which leverages two stage of fine-tuning (uses the large distillation data on stage 1 of fine-tuning and the human-annotated data on stage 2 of fine-tuning) brings the F1 score back to within 1% of the LUKE_{ft} model. Since **$\text{DistilBERT}_{\text{dtft}}$** model has the same model architecture and the same number of parameters as the $\text{DistilBERT}_{\text{ft}}$ model, its inference time is identical: 40ms, i.e. 75x faster than LUKE_{ft}. This makes **$\text{DistilBERT}_{\text{dtft}}$** model applicable for production deployment as it achieves an improved F1 score with high efficiency while requiring less computational resources due to its small size.

6 Conclusion

In this paper, we introduce the *distill-then-fine-tune* method for entity recognition on real world noisy data to deploy our NER model in a limited budget production environment. By generating pseudo-labels using a large teacher model pre-trained on typed text while fine-tuned on noisy speech text to train a smaller student model, we make the student model 75x times faster while reserving 99.09% of its accuracy. These findings demonstrate that our proposed approach is very effective in limited budget scenarios to alleviate the need of human labeling of a large amount of noisy data. In the future, we will explore how to apply knowledge distillation to other tasks (Laskar et al., 2022a,b; Khasanova et al., 2022) containing noisy data.

Ethics Statement

The data used in this research is comprised of individual sentences that do not contain sensitive, personal, or identifying information. Each machine-sampled utterance is labelled by annotators before the utterance is used as part of the training dataset. 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 in these experiments, thereby mitigating this unknown to some extent. Future work should nonetheless strive to improve training data further in this regard.

References

Alan Akbik, Duncan Blythe, and Roland Vollgraf. 2018. Contextual string embeddings for sequence labeling. In *Proceedings of the 27th International Conference on Computational Linguistics, COLING 2018, Santa Fe, New Mexico, USA, August 20-26, 2018*, pages 1638–1649. Association for Computational Linguistics.

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)*, Minneapolis, Minnesota. Association for Computational Linguistics.

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, Online. Association for Computational Linguistics.

Jianping Gou, Baosheng Yu, Stephen J Maybank, and Dacheng Tao. 2021. Knowledge distillation: A survey. *International Journal of Computer Vision*, 129(6):1789–1819.

Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. 2015. Distilling the knowledge in a neural network. *CoRR*, abs/1503.02531.

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional LSTM-CRF models for sequence tagging. *CoRR*, abs/1508.01991.

Elena Khasanova, Pooja Hiranandani, Shayna Gardiner, Cheng Chen, Simon Corston-Oliver, and Xue-Yong Fu. 2022. Developing a production system for Purpose of Call detection in business phone conversations. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Industry Track*, pages 259–267, Hybrid: Seattle, Washington + Online. Association for Computational Linguistics.

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, Hybrid: Seattle, Washington + Online. 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.

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.

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.

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.

Raphael Tang, Yao Lu, Linqing Liu, Lili Mou, Olga Vechtomova, and Jimmy Lin. 2019. Distilling task-specific knowledge from BERT into simple neural networks. *CoRR*, abs/1903.12136.

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.