INTERSPEECH 2020
October 25–29, 2020, Shanghai, China

End-to-end Named Entity Recognition from English Speech

Hemant Yadav[1], *Sreyan Ghosh*[1], *Yi Yu*[2], *Rajiv Ratn Shah*[1]

[1]MIDAS, IIIT-Delhi, India
[2]National Institute of Informatics, Tokyo
hemantya@iiitd.ac.in, gsreyan@gmail.com, yiyu@nii.ac.jp, rajivratn@iiitd.ac.in

Abstract

Named entity recognition (NER) from text has been a widely studied problem and usually extracts semantic information from text. Until now, NER from speech is mostly studied in a two-step pipeline process that includes first applying an automatic speech recognition (ASR) system on an audio sample and then passing the predicted transcript to a NER tagger. In such cases, the error does not propagate from one step to another as both the tasks are not optimized in an end-to-end (E2E) fashion. Recent studies confirm that integrated approaches (*e.g.*, E2E ASR) outperform sequential ones (*e.g.*, phoneme based ASR). In this paper, we introduce a first publicly available NER annotated dataset for English speech and present an E2E approach, which jointly optimizes the ASR and NER tagger components. Experimental results show that the proposed E2E approach outperforms the classical two-step approach. We also discuss how NER from speech can be used to handle out of vocabulary (OOV) words in an ASR system.

Index Terms: End-to-end ASR, named entity recognition, deep learning, out of vocabulary (OOV) words.

1. Introduction

Named entities are phrases that contain the names of persons, organizations, locations, others. For example, the sentence, *T.C.S. CEO Rajesh Gopinathan heads a meeting in their Banglore office*, has the following named entities: [ORG T.C.S.], [PER Rajesh Gopinathan], and [LOC Banglore]. ORG, PER, and LOC represent the organization, person, and location, respectively. In this paper, we focus on these three named entities.

NER is an important task in information extraction systems and very useful in many applications. It has many progress [1, 2, 3] and applications such as in optimizing search engine algorithms [4], classifying content for news providers [5] and recommending content [6]. However, despite NER from speech has many applications such as the privacy concerns in medical recordings (*e.g.*, to mute or hide specific words such as patient names) [7], it has very limited literature.

NER from English speech is done by a classical two-step approach [7]. It consists of first processing the given audio on an ASR system and then feeding the transcribed ASR output to the NER tagger [8, 9] (see Figure 1). Such approaches have several disadvantages, such as existing NER systems are not robust to the noisy output of the ASR, since they are usually designed to process written language. Furthermore, usually, no information corresponding to named entities are used in the ASR system. However, such information could be used to choose better partial hypotheses which are dropped away during the decoding step. As a consequence, even when the decoding goes beyond the 1-best ASR hypothesis for better robustness to ASR errors [10], the search space is pruned without taking into account knowledge of the partial named entities. In all these cases the NER component is trained independently. Thus, the error does not propagate from one step to another in an E2E fashion.

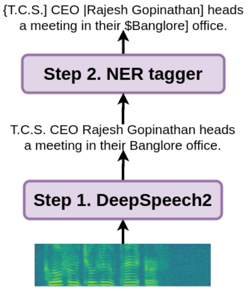
{T.C.S.] CEO |Rajesh Gopinathan] heads
a meeting in their $Banglore] office.
Step 2. NER tagger
T.C.S. CEO Rajesh Gopinathan heads
a meeting in their Banglore office.
Step 1. DeepSpeech2

Figure 1: *Two-step approach for NER from speech.*

To the best of our knowledge, the only work related to E2E NER from speech is done on French datasets [11]. This paper is highly motivated by this work. Additionally, we study the effect of a LM on the E2E NER task. Our major contributions are as follows: (i) we introduce a first publicly available NER annotated dataset[1] for English speech, (ii) we present a state-of-the-art approach for an E2E named entities recognition on the curated dataset, and (iii) we discuss how an E2E system can be used to handle out of vocabulary words in an ASR system.

The paper is structured as follows. Section 2 discusses the related work. Section 3 introduces the dataset and Section 4 describes our methodology. We present our experiments in Section 5. Section 6 discusses how to deal with the OOV words, and Section 7 presents the conclusion and future work.

2. Related work

In the literature many studies [12, 3, 13] focused on NER from textual documents. State-of-the-art (SOTA) NER systems leverage advances in deep learning and recent approaches that take advantage from both word and/or character-level embedding [14, 15, 16]. However, NER from speech is a less studied problem in the research community. Until very recently, the majority of work in recognizing named entities from speech is done using the two-step approach, including the audio de-identification task [7], and the work leveraging OOV information to increase the robustness of NER tagger [17].

Named entity recognition from speech using an E2E approach has a very limited literature, and moreover there is no work on English speech. Recently, Ghannay et al. [11] presented an E2E NER on the French datasets. In their work, they used special symbols to achieve NER tagging capabilities in their E2E approach. The special symbols used by them are:

[1] https://doi.org/10.5281/zenodo.3893954

http://dx.doi.org/10.21437/Interspeech.2020-2482

[, (, {,$, &, %, #,) and] (the first eight symbols to denote the start of 8 different named entities and the last common symbol to denote an end to all of them). Similar to their work, we use three special symbols ('{','|', '$') to recognize three most frequent named entities (names of organization, person, and location) from our English speech dataset. In this paper, we compare the E2E and the two-step approach for the English speech. Furthermore, we study the effect of a LM on the E2E NER task.

3. Dataset

The annotated English dataset we prepare for the NER from speech task is a subset of a combination of Librispeech [18], CommonVoice [19], Tedlium [20] and Voxforge [21]. The recordings are comprised mainly of two domains: Reading English and Ted talks. We refer to this combined data as DATA1, which has around 600,000 files (approximately 1,000 hrs). After an empirical analysis, we found that the majority of files did not have any named entities. To remove these files from the manual annotation step, we use Flair [9, 8] as a NER tagger with 0.9 F1 score. In this way, we have reduced the number of files having NER to 70,769, which are used for manual annotation.

> **Character sequence:** T.C.S. CEO Rajesh Gopinathan heads a meeting in their Banglore office.
> **Character sequence with named entities:** [ORG T.C.S.] CEO [PER Rajesh Gopinathan] heads a meeting in their [LOC Banglore] office.
> **Character sequence with special symbols:** {T.C.S.] CEO |Rajesh Gopinathan] heads a meeting in their $Banglore] office.

Figure 2: *Example of mapping the named entities. '|', '$', '{' denote the start of a named entity and ']' denotes the end.*

Thus, the dataset is prepared into two steps: (i) applying a NER tagger on DATA1 (600,000 files) and (ii) manually annotating the 70,769 files having valid NERs using Doccano[2] [22]. In step 1, we re-train the Flair tagger on the capitalized NER benchmark CoNLL-2003 [23] dataset from scratch. The transcripts at the test time are capitalized, and so is the training data. Furthermore, after an empirical analysis, it was found that a threshold probability of 0.95 rejects the majority of noisy/erroneous named entities from the tagger output. After the Flair NER tagger operation on DATA1, the total number of files is reduced to 70,769 (approx. 150 hrs) from 600,000 (approx. 1,000 hrs). We refer to this reduced data as DATA2. Since DATA1 has audios of different speakers recording the same sentence, DATA2 also has repetitions. To be precise, DATA2 has a total of 31,000 unique sentences, and the rest 39,769 are a repetition of 1238 sentences from 31,000 unique sentences.

In step 2, We manually annotate all the remaining 70,769 files in DATA2 following CoNLL-2003 [23] guidelines. An example of the manually annotated sentence is shown in Figure 2, i.e., character sequence with and without the named entities. To increase the robustness of the model, similar to [24], we asked the annotator to randomly mislabel some tokens as named entities (*e.g.*, annotating the CEO token as [PER] or Banglore token as [ORG]).

The named entities distribution in DATA2 are shown in Table 1, and DATA2 has a total of 38891 unique named entity tokens, as shown in the last row of the Table 1. Furthermore, DATA2 is comprised of 34% Librispeech, 36% CommonVoice, 7% Tedlium, and 23% Voxforge of DATA1.

Table 1: *Category wise distribution of unique and repeated sentences in DATA2.*

Category	Unique	Repetition	Total
Person	24711	20268	44979
Location	11881	7948	19829
Organization	2299	1473	3772
Total	38891	29689	68580

4. Methodology

This section provides information on the E2E approach used in this study, the training method used to train it and the different components used in the decoding step at the test time.

4.1. Language Model

A language model (LM) is a probability distribution over an arbitrary symbol sequences $P(w_1, ..., w_n)$ such that more likely sequences are assigned higher probabilities and vice versa. LMs are frequently used at the decoding step to condition beam search. During the decoding, the top-n candidates are evaluated by conditioning the output of acoustic model with the language model. In this study 4-gram LM is used and, to make the comparison fairer, the LM is trained on the full dataset using the KENLM library [25].

4.2. E2E approach

The problem of NER from speech is to assign a special symbol before and after a named entity to identify it. We approach this problem as a sequence labeling task and use an RNN based Baidu's DeepSpeech2 (DS2) [26] neural architecture to study NER from speech with modifications in the last layer. DS2 is a combination of Convolution Neural Network (CNN) and Recurrent Neural Network (RNN) layers, with a fully connected followed by a softmax layer. The softmax layer outputs the probabilities of the sequence of characters.

Let $X = \{x_1, x_2, ..., x_n\}$ be the input utterances and $Y = \{y_1, y_2, ..., y_n\}$ be the corresponding transcripts. For a given input audio x_1, it is transformed into a sequence of "log-spectrograms of power normalized audio clips, calculated on a 20ms window", and is then fed to the model, which captures the sequential nature of speech. The output l is a sequence of defined set of characters ('A-Z' + ' '). Similar to the work [11], we add special symbols ('|', '$', '{' denote the start of named entities person, location and organization, respectively, and ']' denotes the end of any three named entities) to the pre-defined set of characters to introduce the named entity tagging capabilities in the architecture. An example of a character sequence with and without the special symbols is shown in the Figure 2. To recognize named entities, we modify the DS2 architecture by increasing the shape of the fully connected layer by four to accommodate the extra symbols in the output layer (see Figure 3).

We train the system using the Connectionist temporal classification [27] (CTC) loss (see Figure 4) because CTC loss takes into account all the possible character sequences given the output and the true transcript. Thus, CTC loss maximizes the total probability of all the paths which lead to the true transcript. The model predicts $p(l_t/x_1)$ at each time step.

[2] https://github.com/doccano

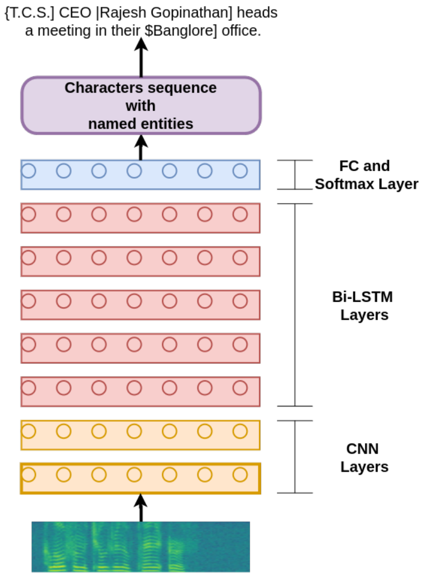
{T.C.S.] CEO |Rajesh Gopinathan] heads a meeting in their $Banglore] office.
Characters sequence with named entities
FC and Softmax Layer
Bi-LSTM Layers
CNN Layers

Figure 3: *E2E approach for the NER task.*

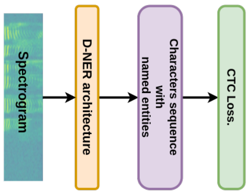

Figure 4: *Training of the E2E architecture.*

At the test time, the output is conditioned on an N-gram LM using prefix beam search decoding [28] and is shown as follows.

$$Q(y) = log(p(l_t/x)) + \alpha * log(pLM(y)) + \beta * wc(y)$$

Where $wc(y)$ is the word count in the predicted transcript. α and β control the contribution of LM and the number of words in the predicted transcript. In this study, we use values 1.96 & 6.0 for α and β, respectively.

To compensate for the limited data for the NER from speech task, DS2 weights are used as a starting point to train the E2E model. We train a standard DS2 architecture on the DATA1 as a baseline. This system achieves a word error rate (WER) of 2.72% on the test set with a 4-gram LM and beam-width equal to 1024. In a similar work on the French dataset [11], they achieved a WER of 19.96%. Better WER is the major reason that our E2E NER system achieves better results compared to [11].

5. Experiments

In this section we discuss the experimental setup of E2E and two-step approaches for NER from speech, the method used for evaluating the two systems and the corresponding results.

5.1. Experimental Setup

All the experiments were carried out on DATA2 prepared in Section 3. Both the dev and test set are created from the 31,000 unique files with a 10% distribution each and the remaining files in DATA2 are used for training.

We experimented with two different approaches for the NER from speech task: E2E and a classical two-step approach. For the E2E approach, experiments were carried out on the model explained in Section 4.2. At the test time, Prefix beam search decoding with a beam-width of 1024 and a 4-gram LM (trained on DATA2) is used. In the case of classical two-step approach, we use Baidu's DS2 as the ASR component and Flair as a NER tagger component. In the two-step approach, audio is first transcribed using the ASR component, and then the output is passed to the Flair tagger as shown in the Figure 1. The Flair tagger used in the two-step approach is explained in the Section 5.2.

5.2. NER Tagger

NER, also known as entity identification, and entity extraction, seeks to locate and classify named entities in text into some predefined labels. For this study, we use Flair as the NER tagger in the two-step approach. The Flair tagger is trained on a combined dataset of DATA2 and CoNLL-2003 [23] and the test set is same as in the case of the E2E approach. The results for the Flair NER tagger on the test set are shown in Table 2.

Table 2: *Precision, Recall and F1 score of the Flair NER tagger used in place of the step 2 in classical two-step approach.*

Category	Precision	Recall	F1
Person	0.84	0.88	0.86
Location	0.87	0.86	0.87
Organization	0.86	0.77	0.81

5.3. Evaluation

Similar to the work [11], we use F1 score [29] for evaluation, which is defined as follows.

$$F_1 = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

Precision is the percentage of named entities that are correct in the predictions compared to true labels and recall is the percentage of ground truth named entities found by the system. A named entity is correct only if it is an exact match to the ground truth entity. We emphasize on using micro average since there is a class imbalance as shown in Table 1.

To consider the case of repeated tags and half-labeled predictions, we slightly modify the precision and recall calculation. For example, if a sentence has two identical named entities, then we collapse them and treat them as one. Secondly, we discard any tags which are half-labeled (e.g., in {T.C.S.] CEO |Rajesh Gopinathan heads a meeting in their $Banglore] office. There is no end label to the PER tag.). Apart from these two special cases, we followed all the standard guidelines for calculating precision and recall.

5.4. Experimental Results

In this section, we report results on the DATA2 dataset. In all the experiments, we use batch normalization and weight decay as regularization. The evaluation metrics we report are precision, recall, and F1 score on the NER task. The reader should have in mind that the half-labeled tags are discarded for all the calculations, as mentioned in the Section 5.3. Pre-trained models and training configurations are available on GitHub[3].

Table 3: *Precision, Recall and F1 score of the two-step and E2E NER from speech, with the LM.*

System	Category	Precision	Recall	F1
Two-step	Person	0.82	0.82	0.82
	Location	0.83	0.79	0.81
	Organization	0.75	0.16	0.27
	Micro average	0.83	0.77	0.80
	macro average	0.80	0.59	0.63
E2E NER	Person	**0.96**	**0.86**	**0.91**
	Location	**0.97**	**0.85**	**0.91**
	Organization	**0.96**	**0.70**	**0.81**
	Micro average	**0.96**	**0.85**	**0.90**
	macro average	**0.96**	**0.80**	**0.87**

Table 3 shows our experimental results on the E2E and two-step approach. The scores are in order of person, location, and organization from better to worse because of the class imbalance present in the DATA2, as shown in Table 1. Furthermore, our results prove that the E2E approach outperforms the classical two-step approach in all the cases and by a significant margin. Moreover, the recall for both the approaches is lower compared to the precision. we think that it is because of the smaller size of dataset for the task. The similar trend of precision and recall is observed in the work on French datasets [11].

We also studied the effect of language model on the F1 scores. LM improves the NER scores with a significant margin, and the same can be inferred from Table 4, which shows an improvement by a factor with LM. Therefore, based on the quantitative analysis, we conclude that the NER results are closely dependent on the language model, and if an LM is trained on much a bigger corpus, then the recall could be increased further.

Lastly, if we look at the Table 2, the F1 scores for detecting a named entity is less compared to the E2E approach as shown in Table 3. Therefore, even if we can get the perfect transcriptions from the ASR component, the best we could do is still less than the E2E approach. The reason could be, in the E2E approach the output and the LM (trained with the named entity tags) are conditioned together in the decoding step. Thus, two sources of information are taken into consideration in the E2E approach. Further analysis is required to make any concrete comments.

Table 4: *E2E NER from speech: Micro-Average scores, with and without LM.*

E2E NER	Precision	Recall	F1
without LM	0.38	0.21	0.27
with LM	**0.96**	**0.85**	**0.90**

6. Handling OOV words in an ASR system

In this section, we discuss how the information on named entities can be used to handle OOV words in an ASR system.

Let us start with some statistics concerning OOV words. We trained the LM on the training data only and found out that the percentage of OOV words in development and test set is around 20% each. Out of those 20% OOV words, around 40% are named entities. If an LM is trained on a significantly bigger corpus, than the share of named entities in the OOV words will increase. Thus most of the OOV words will be the named entities. From this point on, the discussion will be based on just one named entity i.e., person.

My name is Gaurav Yadav.
My name is Modi.
My name is Rajiv.

(a) Standard sentences.

My name is <PER>.

(b) **Modified sentence.**

Figure 5: *Sentences to train the LM.*

So far, from the E2E model, we have the probability of a token being a named entity, and only if we could have the same information from the LM, we can condition it using the modified prefix beam search decoding. This can be achieved, if we replace all the individual person tokens with a <person> token as shown in Figure 5 and train an LM (we call it semantic LM or S-LM) on the updated corpus again. The S-LM would learn the probability of the next token being a <PER> instead of the exact names. Therefore, at the test time, we now have the named entity tag information from the E2E model and the S-LM. At the test time, now we can condition the E2E model output with the S-LM to rank the top n-paths using the modified prefix beam search. The modified "prefix beam search" decoding is such that, it assigns a higher probability to the output of E2E model if the same can be inferred from the S-LM or otherwise. Furthermore, while scoring the top beams, we can condition the named entities on actual person names from a pre-defined dictionary of names.

Since we do not have a concept of an individual word now but a <PER> token. Therefore, the problem of named entities being an OOV word is not anymore. Thus we can now condition the E2E model output heavily on the LM, worrying less about penalizing the OOV words.

7. Conclusion and Future Work

In this paper, we made available a first public English speech dataset with named entities. We also presented a detailed comparison between the E2E and the two-step approaches for NER from speech. Experimental results show that the E2E approach provides better results (F1=**0.906**) compared to the two-step approach (F1=0.803). Additionally, a LM plays an important role to achieve these numbers. It is the first study to recognize named entities in English speech using an E2E approach. To conclude, this study presents promising results in a first attempt to experiment with the E2E approach to recognize named entities and constitutes an interesting start point for future work. In the future, we can study the effect of other loss metrics including: NE-WER (Named Entity Word Error Rate) [30] and ATENE (Automatic Transcription Evaluation for Named Entity) [31], instead of just using WER. Additionally, we can further work on our discussions on handling OOV words in an ASR system.

8. Acknowledgements

Rajiv Ratn Shah is partly supported by the Infosys Center for AI and the Center of Design and New Media[4] at IIIT Delhi.

[3] https://github.com/raotnameh/End-to-end-E2E-Named-Entity-Recognition-from-English-Speech

[4] https://cai.iiitd.ac.in/

9. References

[1] J. P. Chiu and E. Nichols, "Named entity recognition with bidirectional lstm-cnns," *Transactions of the Association for Computational Linguistics*, vol. 4, pp. 357–370, 2016.

[2] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, and C. Dyer, "Neural architectures for named entity recognition," *arXiv preprint arXiv:1603.01360*, 2016.

[3] D. Nadeau and S. Sekine, "A survey of named entity recognition and classification," *Lingvisticae Investigationes*, vol. 30, no. 1, pp. 3–26, 2007.

[4] S. Rüd, M. Ciaramita, J. Müller, and H. Schütze, "Piggyback: Using search engines for robust cross-domain named entity recognition," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*. Association for Computational Linguistics, 2011, pp. 965–975.

[5] G. Kumaran and J. Allan, "Text classification and named entities for new event detection," in *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, 2004, pp. 297–304.

[6] K. Koperski, J. Liang, and N. Roseman, "Content recommendation based on collections of entities," Jul. 18 2017, uS Patent 9,710,556.

[7] I. Cohn, I. Laish, G. Beryozkin, G. Li, I. Shafran, I. Szpektor, T. Hartman, A. Hassidim, and Y. Matias, "Audio de-identification: A new entity recognition task," *arXiv preprint arXiv:1903.07037*, 2019.

[8] A. Akbik, T. Bergmann, and R. Vollgraf, "Pooled contextualized embeddings for named entity recognition," in *NAACL 2019, 2019 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, 2019, p. 724728.

[9] A. Akbik, D. Blythe, and R. Vollgraf, "Contextual string embeddings for sequence labeling," in *COLING 2018, 27th International Conference on Computational Linguistics*, 2018, pp. 1638–1649.

[10] D. Hakkani-Tür, F. Béchet, G. Riccardi, and G. Tur, "Beyond asr 1-best: Using word confusion networks in spoken language understanding," *Computer Speech & Language*, vol. 20, no. 4, pp. 495–514, 2006.

[11] S. Ghannay, A. Caubriere, Y. Esteve, A. Laurent, and E. Morin, "End-to-end named entity extraction from speech," *arXiv preprint arXiv:1805.12045*, 2018.

[12] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," *CoRR*, vol. abs/1810.04805, 2018. [Online]. Available: http://arxiv.org/abs/1810.04805

[13] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," *CoRR*, vol. abs/1802.05365, 2018. [Online]. Available: http://arxiv.org/abs/1802.05365

[14] A. Baevski, S. Edunov, Y. Liu, L. Zettlemoyer, and M. Auli, "Cloze-driven pretraining of self-attention networks," *arXiv preprint arXiv:1903.07785*, 2019.

[15] Y. Jiang, C. Hu, T. Xiao, C. Zhang, and J. Zhu, "Improved differentiable architecture search for language modeling and named entity recognition," 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)*. Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 3585–3590. [Online]. Available: https://www.aclweb.org/anthology/D19-1367

[16] J. Straková, M. Straka, and J. Hajic, "Neural architectures for nested NER through linearization," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, Jul. 2019, pp. 5326–5331. [Online]. Available: https://www.aclweb.org/anthology/P19-1527

[17] C. Parada, M. Dredze, and F. Jelinek, "Oov sensitive named-entity recognition in speech," in *Twelfth Annual Conference of the International Speech Communication Association*, 2011.

[18] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an asr corpus based on public domain audio books," in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 5206–5210.

[19] Wikipedia contributors, "Common voice Wikipedia, the free encyclopedia," 2020, [Online; accessed 14-April-2020]. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Common_Voice&oldid=939008593

[20] A. Rousseau, P. Deléglise, and Y. Estève, "Ted-lium: an automatic speech recognition dedicated corpus." in *Conference on Language Resources and Evaluation (LREC)*, 2012, pp. 125–129.

[21] W. contributors, "Voxforge Wikipedia, the free encyclopedia," 2019, [Online; accessed 14-April-2020]. [Online]. Available: https://en.wikipedia.org/w/index.php?title=VoxForge&oldid=913093799

[22] H. Nakayama, T. Kubo, J. Kamura, Y. Taniguchi, and X. Liang, "doccano: Text annotation tool for human," 2018, software available from https://github.com/doccano/doccano. [Online]. Available: https://github.com/doccano/doccano

[23] E. F. Sang and F. De Meulder, "Introduction to the conll-2003 shared task: Language-independent named entity recognition," *arXiv preprint cs/0306050*, 2003.

[24] B. Frenay and M. Verleysen, "Classification in the presence of label noise: A survey," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 5, pp. 845–869, 2014.

[25] K. Heafield, "Kenlm: Faster and smaller language model queries," in *Proceedings of the sixth workshop on statistical machine translation*. Association for Computational Linguistics, 2011, pp. 187–197.

[26] D. Amodei, S. Ananthanarayanan, R. Anubhai, J. Bai, E. Battenberg, C. Case, J. Casper, B. Catanzaro, Q. Cheng, G. Chen *et al.*, "Deep speech 2: End-to-end speech recognition in english and mandarin," in *International conference on machine learning*, 2016, pp. 173–182.

[27] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in *Proceedings of the 23rd international conference on Machine learning*, 2006, pp. 369–376.

[28] A. Y. Hannun, A. L. Maas, D. Jurafsky, and A. Y. Ng, "First-pass large vocabulary continuous speech recognition using bidirectional recurrent dnns," *arXiv preprint arXiv:1408.2873*, 2014.

[29] Y. Sasaki *et al.*, "The truth of the f-measure. 2007," 2007.

[30] J. S. Garofolo, C.G.Auzanne, and E. M. Voorhees, "The trec spoken document retrieval track: A success story." *NIST SPECIAL PUBLICATION S*, vol. 500, no. 246, pp. 107–130, 2000.

[31] M. A. B. Jannet, O. Galibert, M. Adda-Decker, and S. Rosset, "How to evaluate asr output for named entity recognition?" in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.