Deep Learning End-to-End Automatic Speech Recognition for Spanish in Gong

**Abstract:** E2E ASR approach greatly simplifies the pipeline for building ASR systems. Transformer based models architectures that accepts acoustic-features into Encoder and produce language-tokens by Decoder have a great potential to become SOTA models for ASR task. We applied this architecture to Spanish language based on publicly available datasets and Gong’s Spanish dataset. Studied the achievable performance in terms of WER and inference time, and tuned the model for Gong data.

1. Introduction

ASR – define the task

Metrics: WER and Inference-time

Manual Approaches

Available Automatic Approaches

Data Sources

Automatic Speech Recognition (ASR, aka Speech-to-Text), is historically comprised of two parts – an Acoustic Model, mapping speech to phonemes, and a Language Model, mapping phonemes to words.

“However, current systems lean heavily on the scaffolding of complicated legacy architectures that grew up around traditional techniques, including hidden Markov models (HMMs), Gaussian mixture models (GMMs), hybrid HMM/deep neural network (DNN) systems, and sequence discriminative training methods [2]. These systems also require hand-made pronunciation dictionaries based on linguistic assumptions, extra training steps to derive context-dependent phonetic models, and text preprocessing such as tokenization for languages without explicit word boundaries. Consequently, it is quite difficult for non-experts to develop ASR systems for new applications, especially for new languages”

“There are several types of endto-end architecture for ASR such as connectionist temporal classification (CTC) [3], recurrent neural network (RNN) transducer [4], attention-based encoder decoder [5], and their hybrid models”

While in the fields of Computer Vision and NLP, deep learning models outperform other approaches by large margins, in the field of ASR the vast majority of commercial systems don’t use deep learning. In recent years, there have been reports on success in applying end-to-end (e2e) deep learning architectures to transcribe speech. E2E DL platforms hold the promise of better adaptation to accents or new languages, making the use of phonetic lexicons obsolete. Promising directions are Nvidia’s Jasper, Facebook’s Wav2Letter, Mazzila’s DeepSpeach and ESPnet.

Related works

Transformer based Sequence-to-Sequence models showed “surprising superiority” in multiple ASR benchmarks in comparison with RNN”(see ref 2. “A COMPARATIVE STUDY ON TRANSFORMER VS RNN IN SPEECH APPLICATIONS”, S. Karita et al. 2019). The authors of this paper created a kaldi-stile reproducible recipes for Open Source toolkit ESPnet. We based our experiments on this toolkit.

“One of the major difficulties when applying Transformer to speech applications is that it requires more complex configurations(e.g., optimizer, network structure, data augmentation) than the conventional RNN based models”

The contributions of this work are …

* We conducted a large scale comparative study of model performance on publicly available Spanish Datasets
* We performed hyperparameters search for each dataset (see [“Training Tips for the Transformer Model”, M. Popel et al.])
* We developed reproducible pipeline for experiments and following productization
* We tuned the model for Gong’s dataset

1. ASR E2E Sequence to Sequence Pipeline

We use ESPnet framework and their implementation of Hybrid CTC/attention based ASR

[======================= from [https://github.com/espnet/espnet#asr-results](https://github.com/espnet/espnet" \l "asr-results)

ASR: Automatic Speech Recognition

* **State-of-the-art performance** in several ASR benchmarks (comparable/superior to hybrid DNN/HMM and CTC)
* **Hybrid CTC/attention** based end-to-end ASR
  + Fast/accurate training with CTC/attention multitask training
  + CTC/attention joint decoding to boost monotonic alignment decoding
  + Encoder: VGG-like CNN + BiRNN (LSTM/GRU), sub-sampling BiRNN (LSTM/GRU) or Transformer
* Attention: Dot product, location-aware attention, variants of multihead
* Incorporate RNNLM/LSTMLM/TransformerLM/N-gram trained only with text data
* Batch GPU decoding
* **Transducer** based end-to-end ASR
  + Available: RNN-Transducer, Transformer-Transducer, mixed Transformer/RNN-Transducer
  + Also support: attention mechanism (RNN-decoder), pre-init w/ LM (RNN-decoder), VGG-Transformer (encoder)
* CTC forced alignment

============================ from [https://github.com/espnet/espnet#asr-results](https://github.com/espnet/espnet" \l "asr-results)

TBD: add a section describing Wav2Letter, Jasper and simplified Jasper : convolution layers, residual connections, size of the model etc.

S2S model consists of two neural networks: Encoder and Decoder

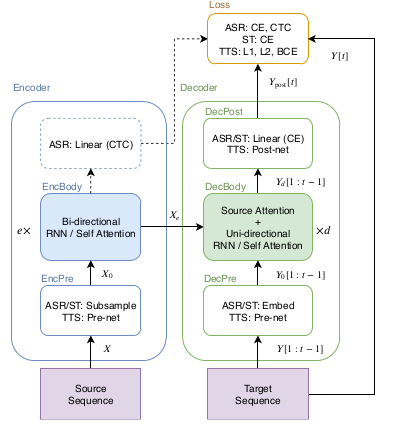


Fig. 1. Sequence to Sequence architecture in speech applications. TBD: redraw for ASR only

Look at the Linear CTC block in the Encoder – this looks like it outputs the char-probabilities vectors per each time-frame. The question is how is it related to the intermediate sequence? “For ASR, an encoded sequence can also be used for source-level frame-wise prediction using connectionist temporal classification (CTC)”

“That attention mechanism emits source frame-wise weights to sum the encoded source frames as a target frame-wise vector to be transformed with the prefix . We refer to this type of attention as “encoder-decoder attention”.”

“Transformer learns sequential information via a self-attention mechanism instead of the recurrent connection employed in RNN.”

“Transformer learns sequential information via a self-attention mechanism instead of the recurrent connection employed in RNN”

“In our ASR framework, the S2S predicts a target sequence Y of characters or SentencePiece [“SentencePiece: A simple and language independent subword tokenizer and detokenizer for Neural Text Processing”, T. Kudo, J. Richardson] from an input sequence of logmel filterbank speech features”.

* 1. Attention Layers in Transformer

“Transformer consists of multiple dot-attention layers:

,

where and are inputs for this attention layer, is the number of feature dimensions, is the length of ,and is the length of and . We refer to as the attention matrix. Vaswani et al., in their original paper, considered these inputs to be query and set of key-value pairs respectively.

”

“In addition, to allow the model to deal with multiple attentions in parallel, Vaswani et al. extended this attention layer to multi-head attention (MHA):

,

,

where

”

* 1. Self-Attention Encoder
  2. Self-Attention Decoder
  3. Encoder architecture

“The source X in ASR is represented as a sequence of 83-dim log-mel filterbank frames with pitch features [“A pitch extraction algorithm tuned for automatic speech recognition”, P. Ghahremani et al.]. First, EncPre(·) trans-forms the source sequence X into a subsampled sequence “Advances in Joint CTC-Attention based End-to-End Speech Recognition with a Deep CNN Encoder and RNN-LM” T. Hori et al.by using two-layer CNN with 256 channels, stride size 2 and kernel size 3 in [“SPEECH-TRANSFORMER: A NO-RECURRENCE SEQUENCE-TO-SEQUENCE MODEL FOR SPEECH RECOGNITION”, L. Dong et al.], or VGG-like max pooling in [“Advances in Joint CTC-Attention based End-to-End Speech Recognition with a Deep CNN Encoder and RNN-LM” T. Hori et al.], where is the length of the output sequence of the CNN. This CNN corresponds to EncPre(·) in Eq. (1). Then, EncBody(·) transforms sub att into a sequence of encoded features for the CTC and decoder networks.”

* 1. Decoder architecture

“The decoder network receives the encoded sequence and the prefix of a target sequence of token IDs: characters or SentencePiece [“SentencePiece: A simple and language independent subword tokenizer and detokenizer for Neural Text Processing”, T. Kudo, J. Richardson]. First, DecPre(·) in Eq. (3) embeds the tokens into learnable vectors. Next, DecBody(·) and single-linear layer DecPost(·) predicts the posterior distribution of the next token prediction given and .”

* 1. Training and Decoding

“During ASR training, both the decoder and the CTC module predict the frame-wise posterior distribution of Y given corresponding source X: and , respectively. We simply use the weighted sum of those negative log likelihood values:

, (14)

where α is a hyperparameter.

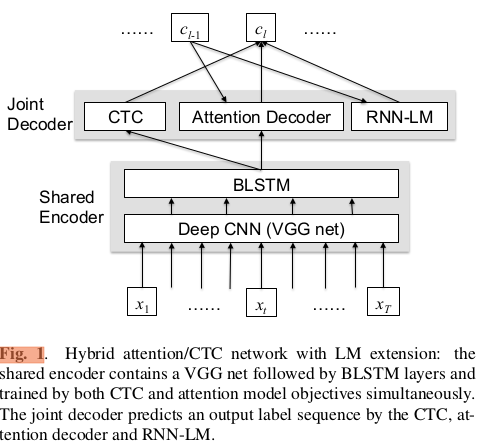
In the decoding stage, the decoder predicts the next token given the speech feature X and the previous predicted tokens using beam search, which combines the scores of ATT, CTC and the RNN language model (LM) [“Recurrent neural network based language mode”, T. Mikolov et al.] as follows:

, (15)

where is a set of hypotheses of the target sequence, and , are

hyperparameters.

TBD: use pytorch and tensorboard to visualize network topology



1. Experiments

TBD:

1. Data Sets

TBD: put a table presenting all used datasets and their characteristics,

Inspect the datasets: the audios, the transcripts

Common Voice contains 148375 utterances, 120 hours, # speakers

Gong Unsupervised

1. Parameters and Optimizations

TBD:

elayers: 12

dlayers: 6

dunits: 2048, eunits: 2048 (feed-forward dimentions)

adim: 512 (attention dimention)

aheads: 8 (number of attention heads)

accum\_grad: 4 see [“Scaling Neural Machine Translation”], <https://github.com/pytorch/fairseq/blob/master/examples/scaling_nmt/README.md>

see [“SPEECH-TRANSFORMER: A NO-RECURRENCE SEQUENCE-TO-SEQUENCE MODEL FOR SPEECH RECOGNITION”, L. Dong et al.] for the selection of the following params

dropout-rate: 0.1

transformer-lr: 10.0

transformer-warmup-steps: 25000

1. Model Tuning for Gong Data

TBD:

1. Lab Journal / Training Details

TBD:

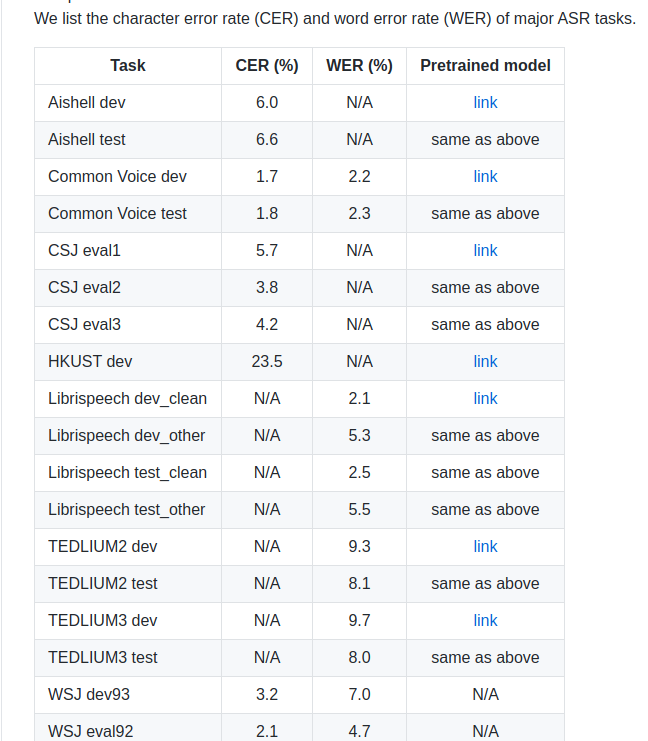
ESPnet tutorial: <https://hackmd.io/s/rJ6TDZPeQ>

1. Results

TBD:

Publicly known results

[https://github.com/espnet/espnet#asr-results](https://github.com/espnet/espnet" \l "asr-results)



1. Discussion

TBD:

1. Conclusion and Future Work

In this paper, we evaluated...

Future work:

1) make comparison with other publicly available frameworks and network architectures in terms of WER and inference time

2) Improve the inference time based on ideas from “STREAMING AUTOMATIC SPEECH RECOGNITION WITH THE TRANSFORMER MODEL”, N. Moritz et al. and “Triggered attention for end-to-end speech recognition” N. Moritz

or find others ideas.

1. Typographical style

1. Figures, supplementary materials, and tables

4.1 Figures

Figures should be included directly in the document. All illustrations must be numbered consecutively (i.e., not by section) with Arabic numbers. The size of a figure should be commensurate with the amount and value of the information conveyed by the figure.

Authors must use one image file per figure. Figures must be inserted as objects that are fixed and move with the text, not as floating objects. Figures should never be placed in a table environment, embedded inside the text, or included within a list. All the figures should be centered. No part of a figure should go beyond the typing area. Place figures as closely as possible to where they are mentioned in the text. Figures should be numbered consecutively in the order of appearance and citation in the text. Be sure to cite every figure.

The abbreviation “Fig.” for figure should appear first followed by the figure number and a period.

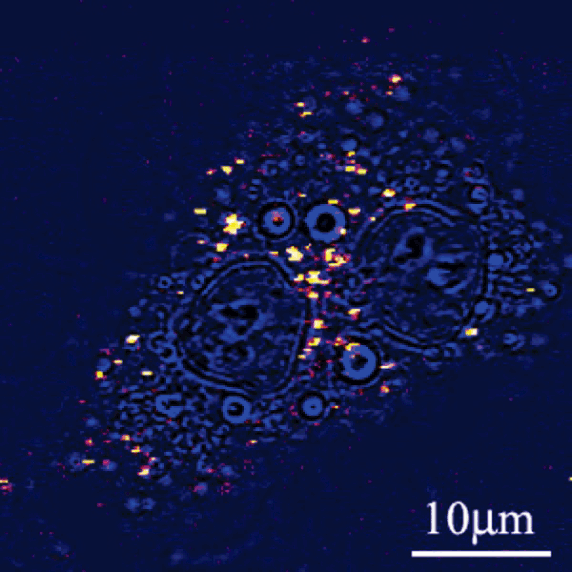


Fig. 1. Sample caption (Ref. [4], Fig. 2).

4.3 Tables

Tables should be centered and numbered consecutively. Authors must use Word’s Table editor to insert tables. Authors must not import tables from Excel. All content for each table should be in a single Word table (do not split content for a single table across multiple Word tables). Tables should use horizontal lines to delimit the top and bottom of the table and column headings. Detailed explanations or table footnotes should be typed directly beneath the table, but not in a table cell. Table footnote labels should be alphabetical; numbers or special characters are not permitted. Position tables as closely as possible to where they are mentioned in the main text.

Table 2. Optical Constants of Thin Films of Materials*a*

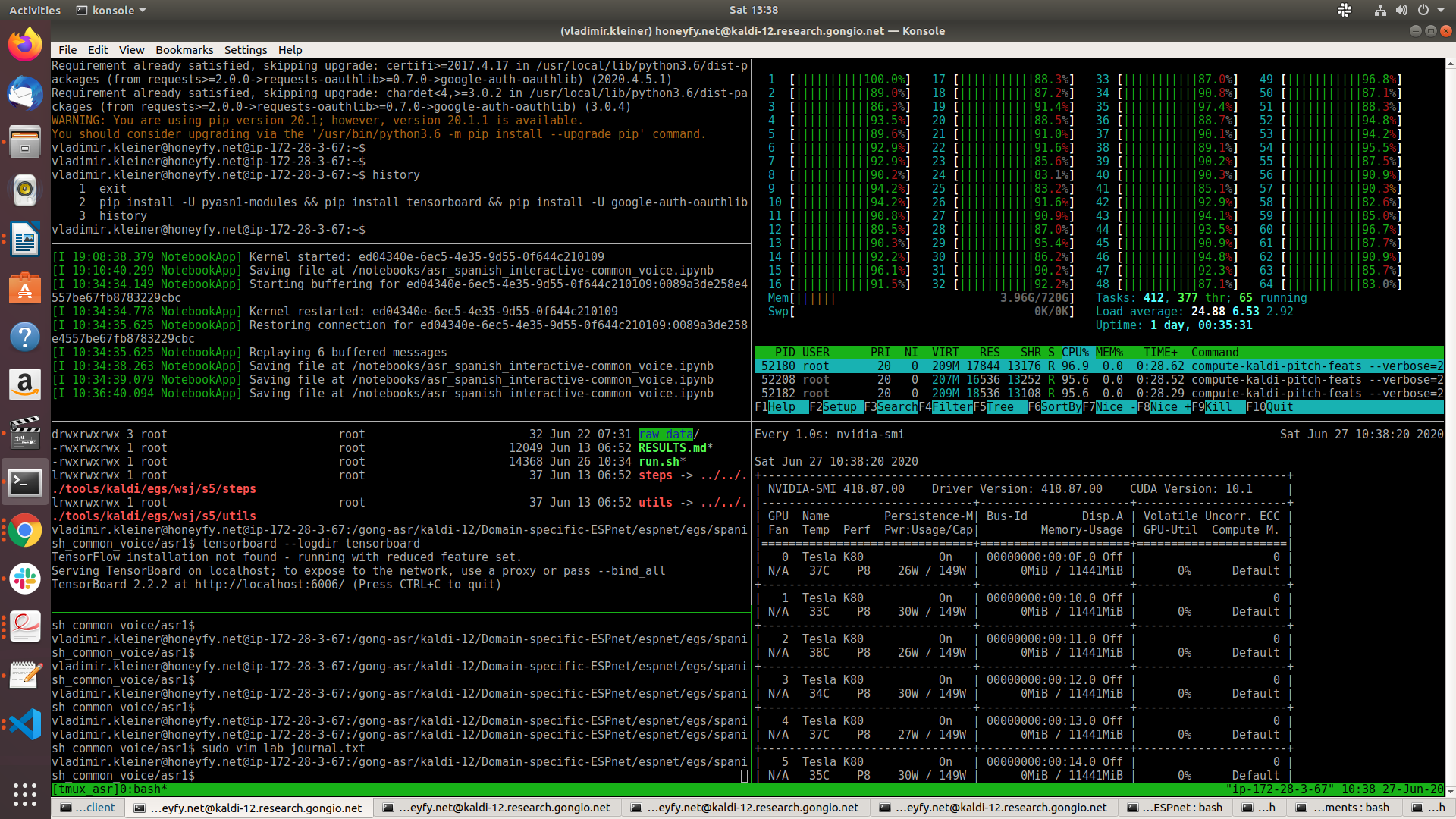
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 83.4 nm | |  | 121.6 nm | |
| Material | n | K |  | n | k |
| Ir | 1.182 | 0.865 |  | 1.450 | 1.040 |
| MgF2 | 1.584 | 0.487 |  | 1.682 | 0.0627 |
| Al | 0.09874 | 0.1915 |  | 0.0424 | 1.137 |
| Mo | 0.98 | 1.08 |  | 0.78 | 1.03 |
| C | 1.16 | 1.29 |  | 1.85 | 1.10 |

*a*From Appl. Opt. **40**, 1128 (2001).

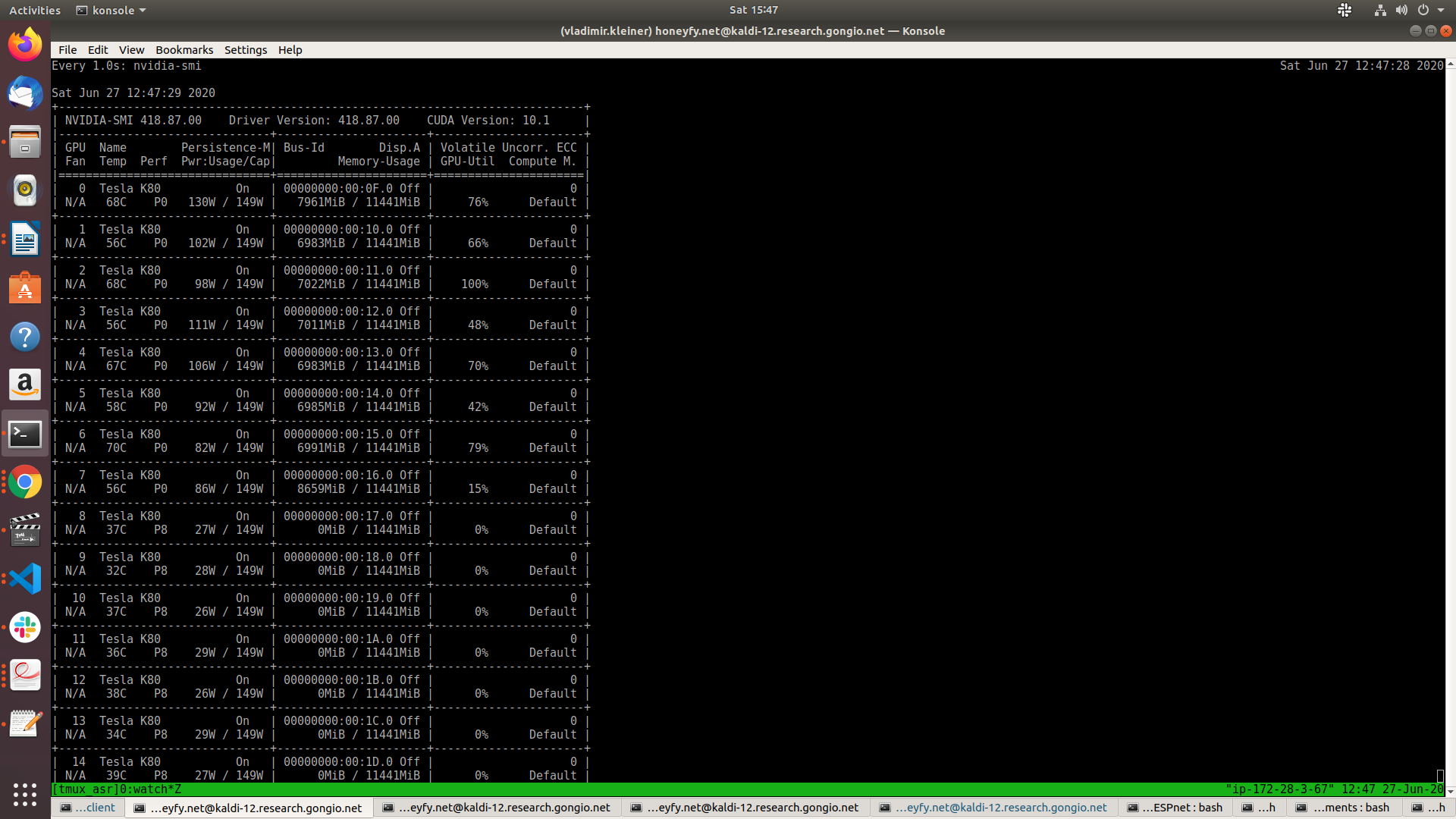
1. Funding, acknowledgments, and disclosures

5.1 Funding

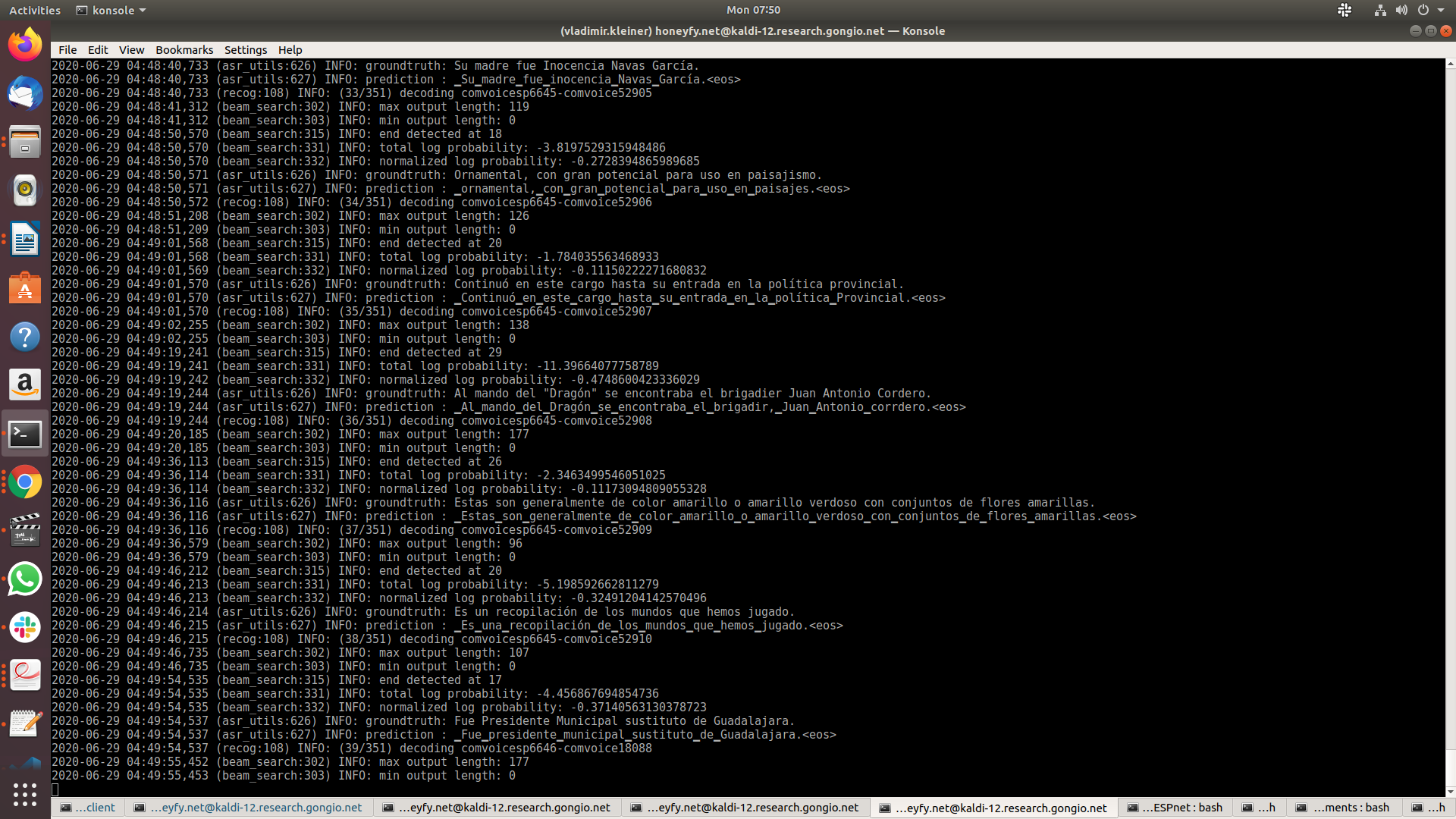
Content in the funding section will be generated entirely from details submitted to Prism. Authors may add placeholder text in the manuscript to assess length, but any text added to this section in the manuscript will be replaced during production and will display official funder names along with any grant numbers provided. If additional details about a funder are required, they may be added to the Acknowledgments, even if this duplicates information in the funding section. See the example below in Acknowledgements.



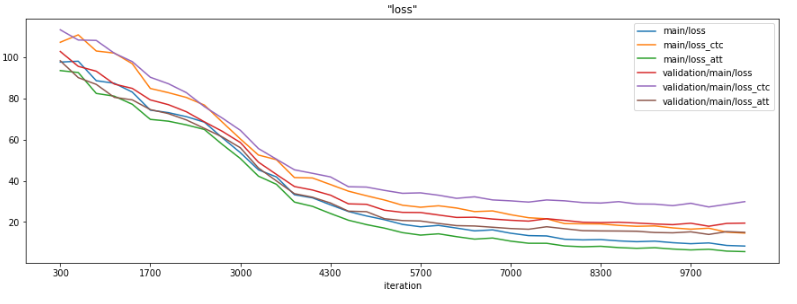
Fd’;glkdf’h;gl



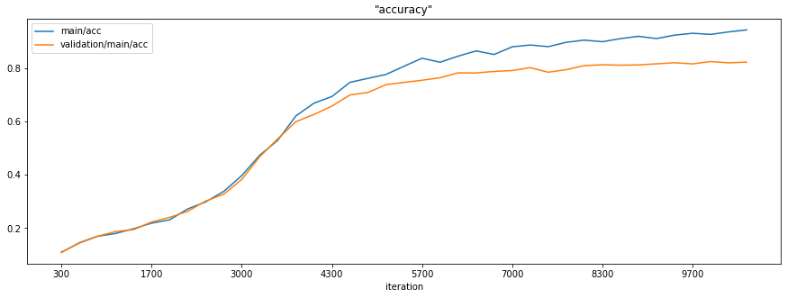
s’;’;lfdgk’hd;lfgkh’fdglk



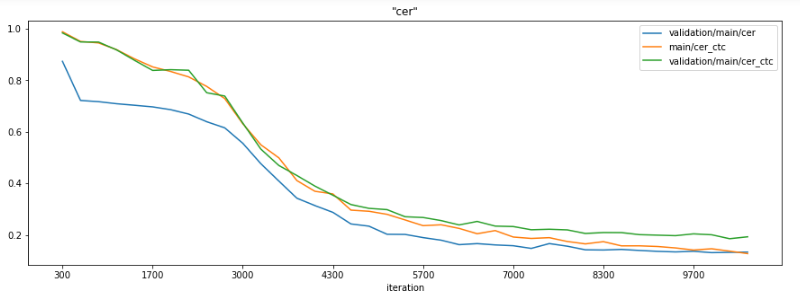
fg’dfg;lhkd’f;lghkdf’gh;



COMMON VOICE



f;lkdf’g;hlkdf’gh;lk



dfg’;h\ldf\g’hl

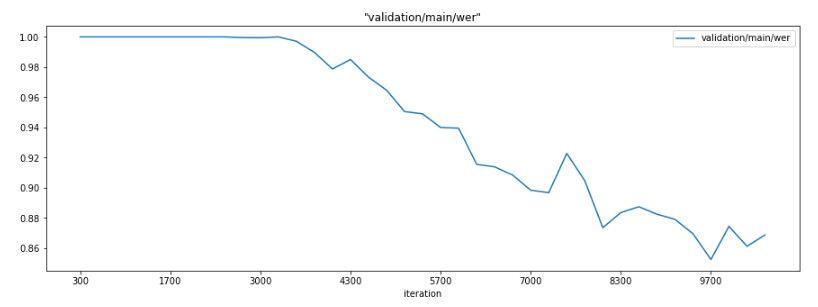
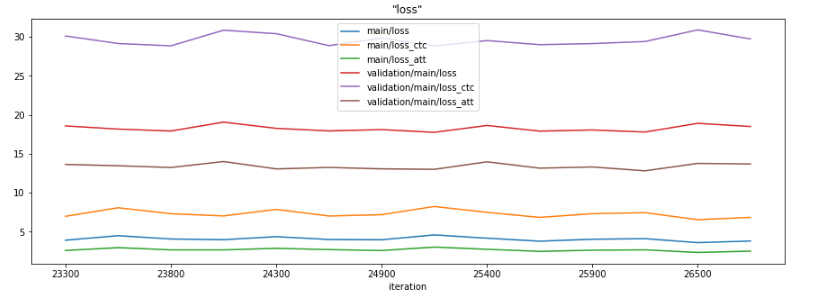
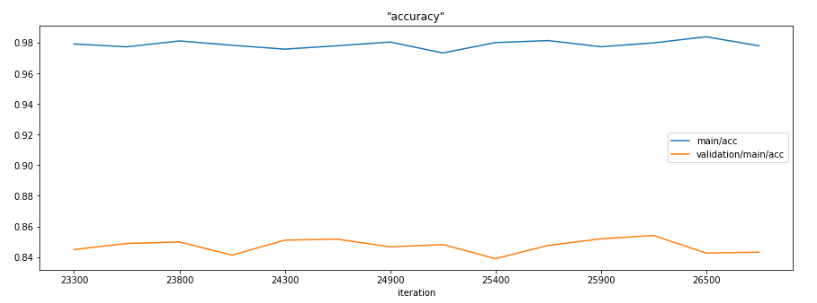
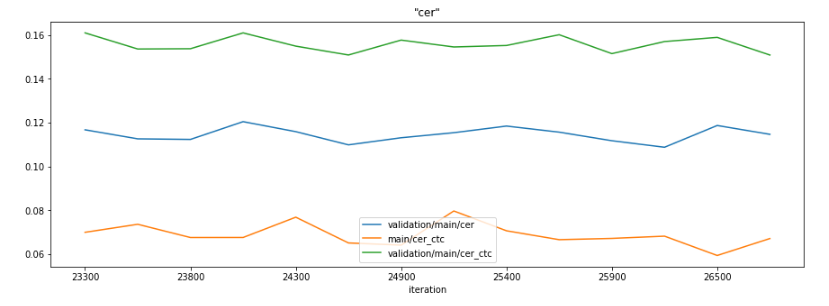


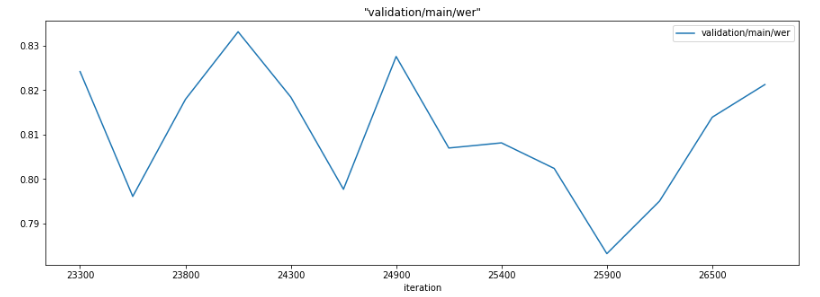
Fig. 2. XYZ.

d\fg;’hd\fgh

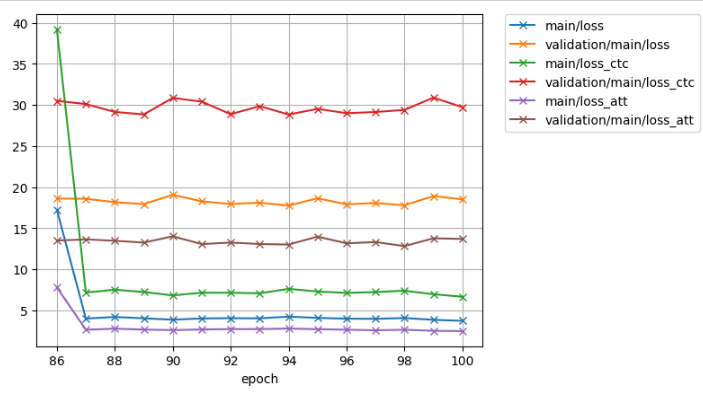


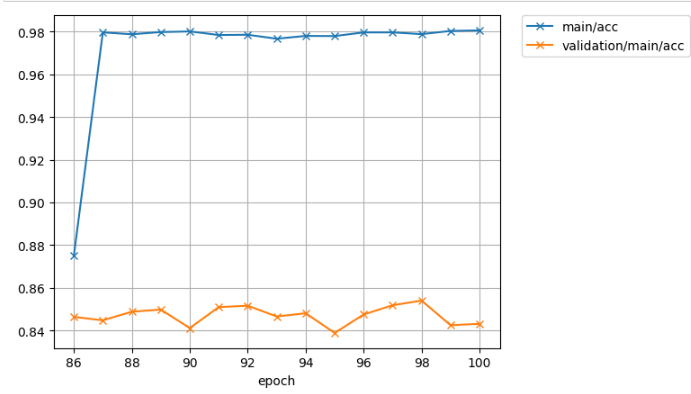
dfgh;dlfghd



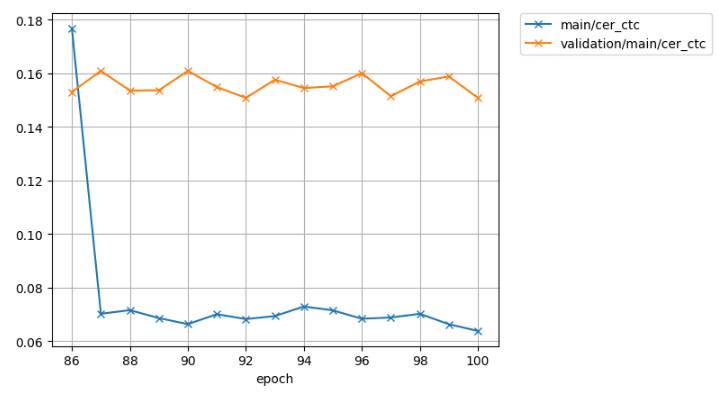


dfghdfghd

dsfgsdfgsdd



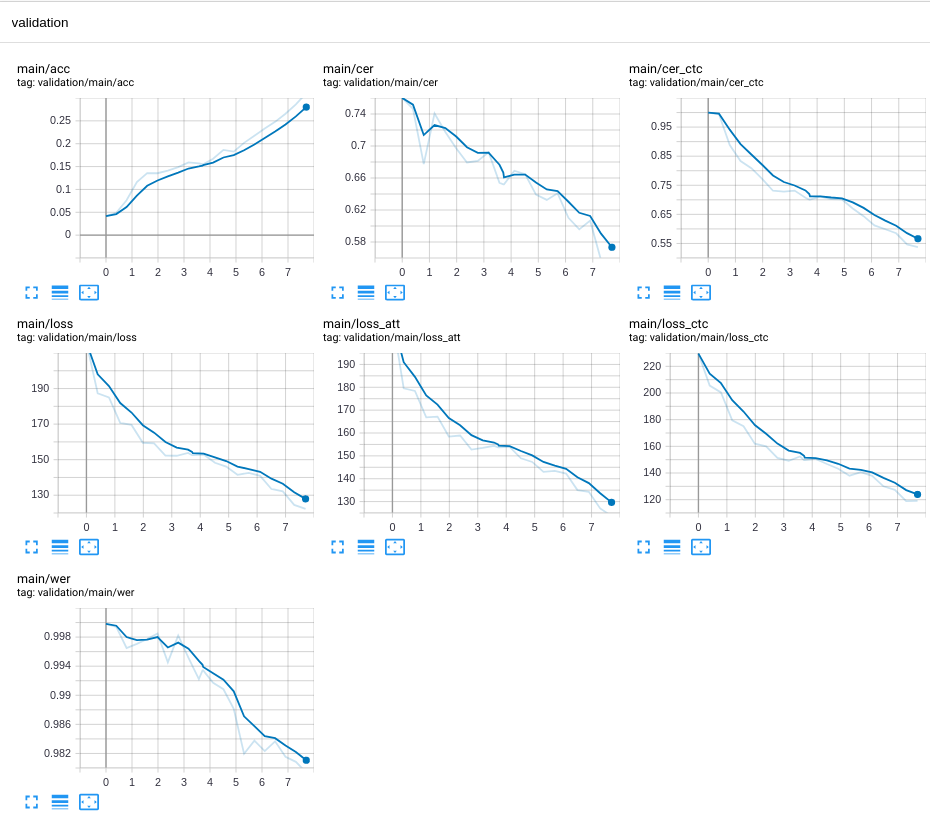
dsfgsdfgdsfg

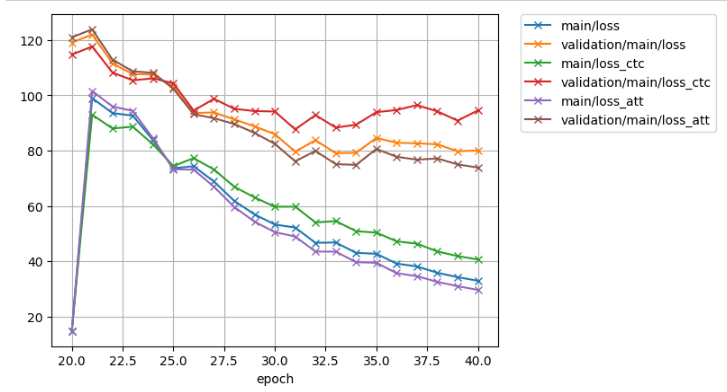


There are no signs of overfitting in spanish common voice after 100 epochs

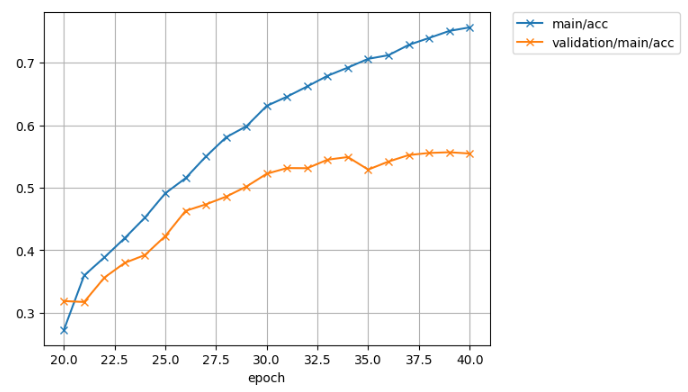
Gong unsupervised

second 10 epochs:

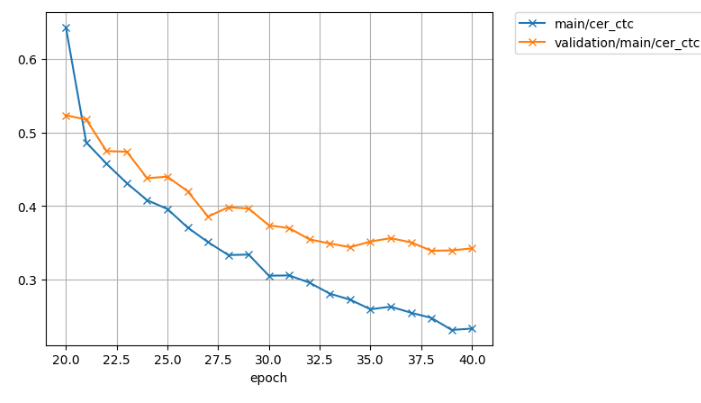


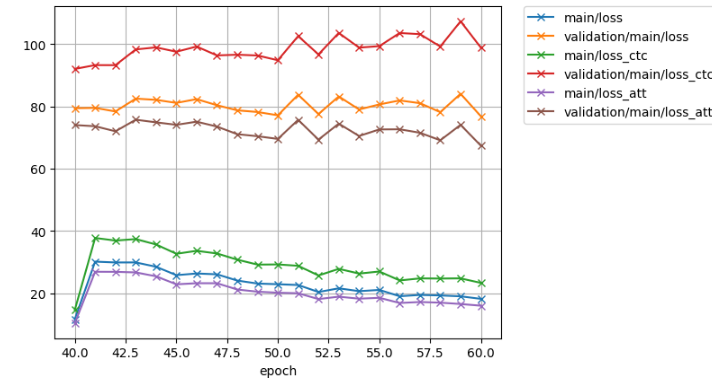


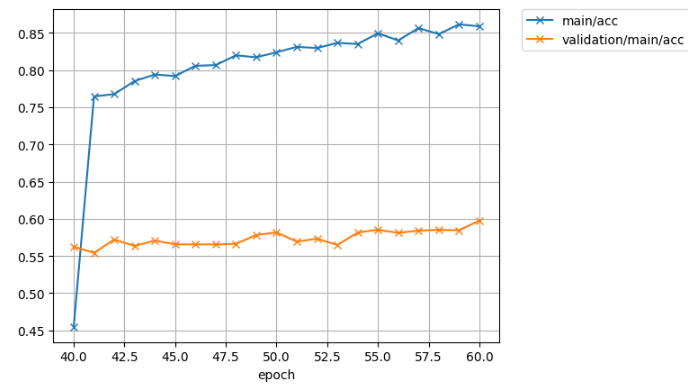
d;flgkd;flgk



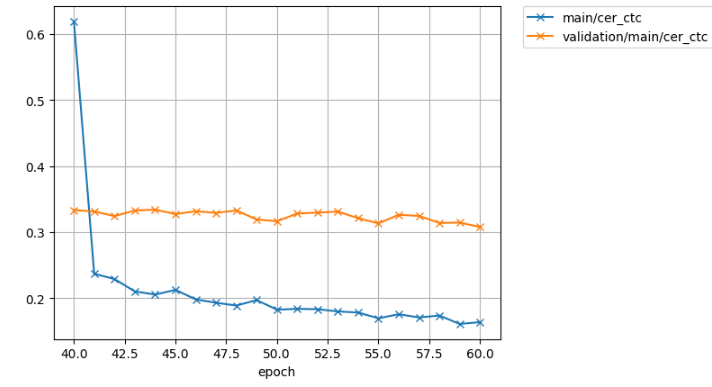
sdfgsdfgsdfg

dfgsdfgsdfgsdfgsdg

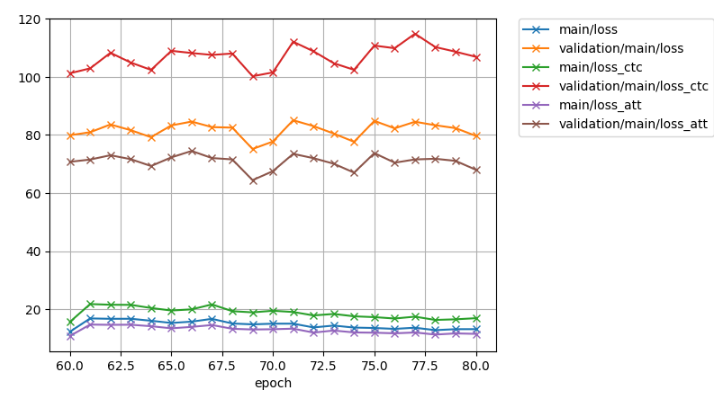
fsd’;lgskd’f;glskdf’g

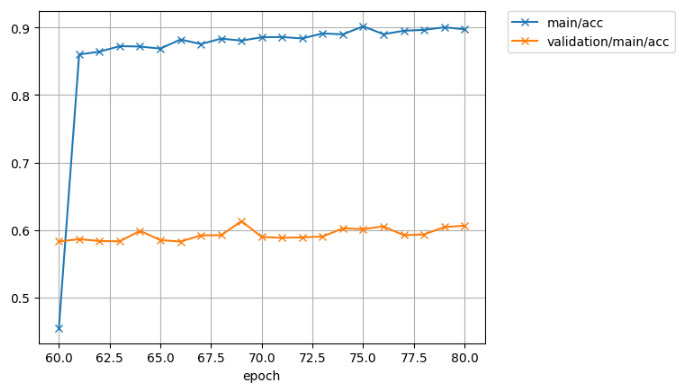


sd’flgsd’f;gls

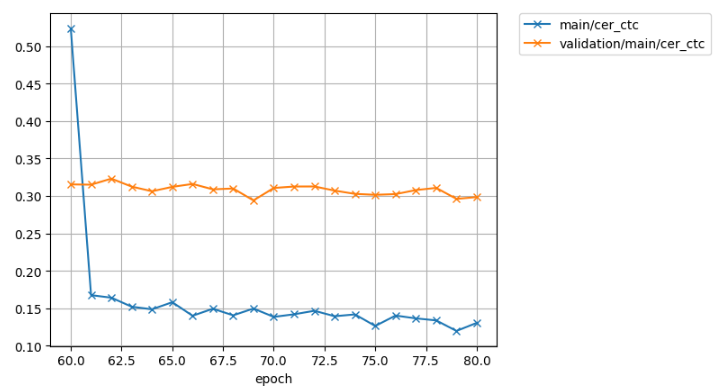


sd’flgks’dfg;l

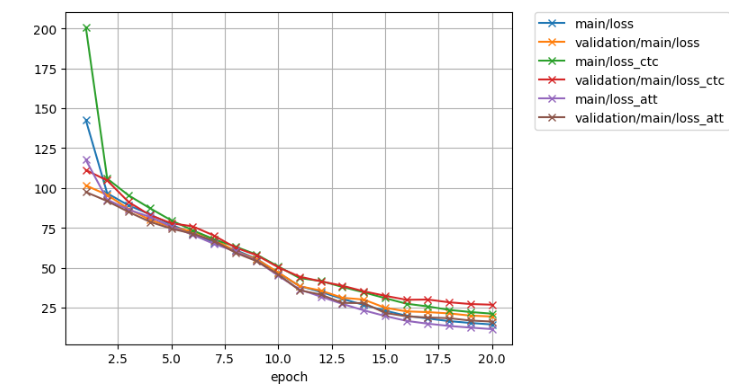
sdf’gsd’;gflsk



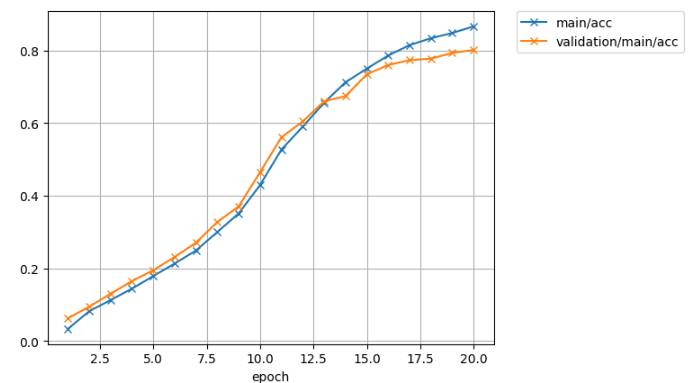
dfsg’sld’fgs;lk



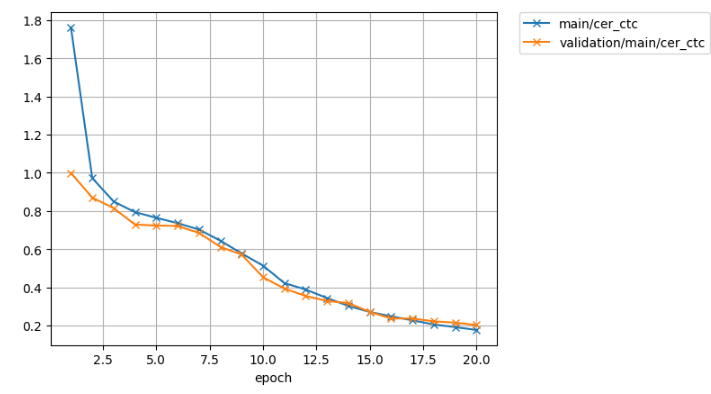
COMMON VOICE with SubWords from Gong-Unsupervised



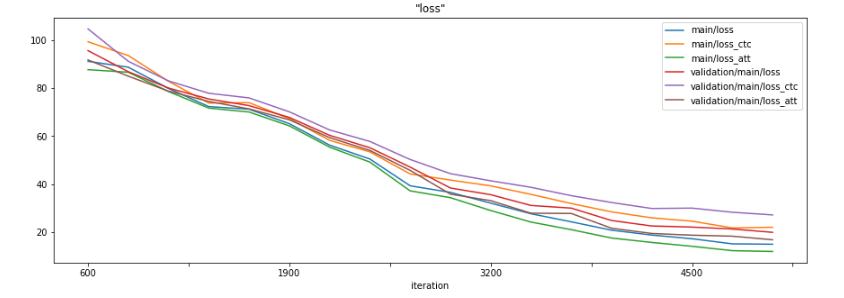
dsf’;gskd’fg



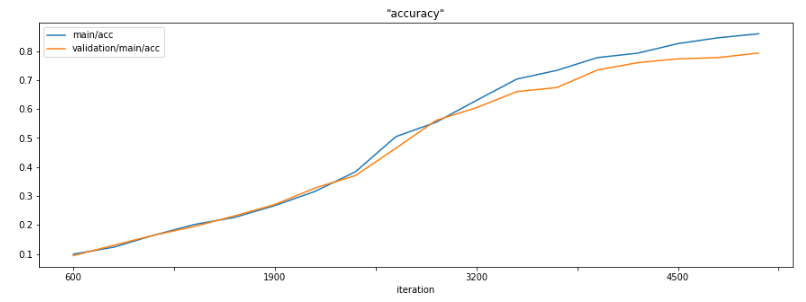
sdfgksd’fg;



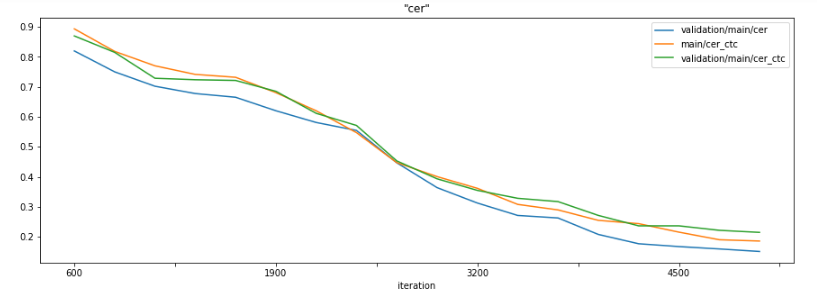
sdfgs’d;flgk



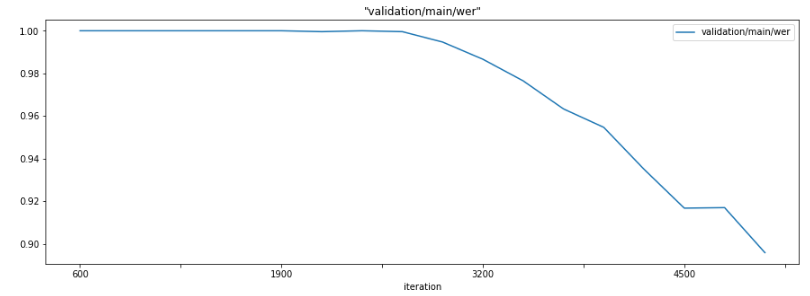
sd’f;glsdk’fg;l



s’df;lgks’df;gs



sdf’;lgksd’f;glk



References

1. “Attention Is All You Need”, A. Vaswani et al.
2. “The Annotated Transformer”, A. Rush, et al. (<http://nlp.seas.harvard.edu/2018/04/03/attention.html>)
3. “A COMPARATIVE STUDY ON TRANSFORMER VS RNN IN SPEECH APPLICATIONS”, S. Karita et al. (the authors are contributed the Transformer-based ASR into ESPnet)
4. “End-to-end speech recognition with word-based rnn language models”, T. Hori et al. 2018
5. “Improving Transformer-based End-to-End Speech Recognition with Connectionist Temporal Classification and Language Model Integration”, S. Karita et al.
6. “JOINT CTC-ATTENTION BASED END-TO-END SPEECH RECOGNITION USING MULTI-TASK LEARNING”, S. Kim et al.
7. “Joint CTC/attention decoding for end-to-end speech recognition”, T. Hori et al.
8. “Advances in Joint CTC-Attention based End-to-End Speech Recognition with a Deep CNN Encoder and RNN-LM” T. Hori et al.
9. “Effective Approaches to Attention-based Neural Machine Translation”, M Luong et al (2015)
10. “The Challenges of using Transformers in ASR”, D. Raj (<https://desh2608.github.io/2020-01-08-transformer-asr/>)
11. “Neural Machine Translation of Rare Words with Subword Unit”, R. Sennrich et al. (introduced BPE)
12. “TRANSFORMER-BASED ACOUSTIC MODELING FOR HYBRID SPEECH RECOGNITION”, Y Wang, et al. (“Though compared with the sequence-to-sequence or neural transducer architecture, the hybrid approach is admittedly less appealing as it is not end-to-end trained, it is still the best performing system for authors’ practical problems. It also has the advantage that it can be easily integrated with other knowledge sources (e.g., personalized lexicon) that may not be available during training. In this work, we aim to leverage the transformer to improve hybrid acoustic modeling”)
13. “Subword Regularization: Improving Neural Network Translation Models with Multiple Subword Candidates”, T.Kudo.
14. “SPEECH-TRANSFORMER: A NO-RECURRENCE SEQUENCE-TO-SEQUENCE MODEL FOR SPEECH RECOGNITION”, L. Dong et al.
15. “Tensor2Tensor for neural machine translation”, A. Vaswani et al.
16. “The Kaldi speech recognition toolkit”, D. Povey et al.
17. “Training Tips for the Transformer Model”, M. Popel et al.
18. “Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks”, A.Graves et al.
19. “A pitch extraction algorithm tuned for automatic speech recognition”, P. Ghahremani et al.
20. “STREAMING AUTOMATIC SPEECH RECOGNITION WITH THE TRANSFORMER MODEL”, N. Moritz et al.
21. “SentencePiece: A simple and language independent subword tokenizer and detokenizer for Neural Text Processing”, T. Kudo, J. Richardson
22. “Recurrent neural network based language mode”, T. Mikolov et al.
23. “Scaling Neural Machine Translation”, M. Ott et al.
24. “STREAMING AUTOMATIC SPEECH RECOGNITION WITH THE TRANSFORMER MODEL” N. Moritz
25. “Triggered attention for end-to-end speech recognition” N. Moritz
26. “Speech recognition for medical conversations” CC Chiu et al. (ASR with noisy transcripts)