INF642 : Socio-emotional Embodied Conversational Agents

Lab 3: Evaluation of the model

29/01/2020 – 1:30 pm

Note: The submission deadline is: 03/02/2020, 11:59 pm. You need to submit a report, explaining every step of your work. Add all the deliverables in a ZIP file and save it with your first and last name.

Objective: The objective of this lab session is to continue working on your seq2seq model, train it this time with cleaned data.

Part 1

Teacher Forcing

Teacher Forcing is a fast and effective way to train your RNN that uses output from previous steps as input to the current step. The main gain of teacher forcing is in the computational training and minimizing the loss landscape.

However, during inference time, the model can result with a limited behavior, as all of you might have noticed, which can lead to poor prediction performance as the RNN's conditioning context (sequence of previously generated samples) diverge from the sequences seen during training.

Exposure Bias Problem [4]

The previously described problem is called the 'Exposure Bias' problem. The problem is that the model never depended on its own errors during training, it only depended on the ground truth provided. During inference, the model is not provided with the ground truth, and it only depends on the previous step, and therefore it only depends on itself.

This problem only arises when the model results in a bad output at the previous timestep, which will affect all the future time step predictions. The model will simply end up in a completely different state space from where it has seen and trained on during training phase.

Therefore, this problem causes a discrepancy between how the model is trained and how it runs in inference.

Solutions – Extensions to Teacher Forcing

To address this limitation, there are number of approaches that you can use:

Beam Search [2][6]:

Makes it possible to optimize during training time through the inference procedure. It uses a differentiable decoder which can be used to efficiently make the predictions without exposure bias.

Curriculum Learning (Scheduled Sampling) [1]

Involves gradually changing the reliance of the model from entirely being dependent on the ground truth being provided to it, to depending on itself (learning from its own errors, i.e. its own previous timesteps). This approach consists of randomly choosing where to sample from during training: either from the ground truth, or the model itself.

Parallel Scheduled Sampling [3]

Consists of generating conditioning tokens for all timesteps in parallel (better to run this method on GPUs and TPUs), simultaneously, and in multiple passes.

Professor Forcing [5]

Was developed to improve long-term sequence sampling from recurrent networks while avoiding Exposure Bias. This approach makes us of the generative adversarial networks (GANs) to match the two distributions over sequences: the one observed in teacher forcing mode, and the one observed in testing/inference mode.

In this first part of the lab, you need to do a research on these different approaches, understand each, explain all of them in your report (+ the pros and cons for each), and **choose one** for your model. Justify your choice in your report.

References

[1] S. Bengio, O. Vinyals, N. Jaitly, and N. Shazeer. Scheduled sampling for sequence prediction with recurrent neural networks (2015), NeurIPS 2015.

[2] R. Collobert, A. Hannun, and G. Synnaeve. A Fully Differentiable Beam Search Decoder (2019), ICML 2019.

[3] D. Duckworth, A. Neelakantan, B. Goodrich, L. Kaiser, and S. Bengio. Parallel Scheduled Sampling (2019), arXiv.

[4] T. He, J. Zhang, Z. Zhou, and J. Glass. Quantifying Exposure Bias for Neural Language Generation (2019), arXiv.

[5] A. Lamb, A. Goyal, Y. Zhang, S. Zhang, A. Courville, and Y. Bengio. Professor Forcing: A New Algorithm for Training Recurrent Networks (2016), NeurIPS 2016. https://arxiv.org/abs/1610.09038

[6] S. Wiseman, and A. Rush. Sequence-to-Sequence Learning as Beam-Search Optimization (2016), EMNLP 2016.

Part 2

The second part of this lab consists of training your model with one of the previously mentioned approaches. Please use the features that are available in the following links:

Batch 1: https://drive.google.com/file/d/1UhmKjlEQo3enQFLLg4AhPKUoTuCiEi2D/view?usp=sharing

Batch 2: https://drive.google.com/file/d/1HEVJBxGbhGomIJLD9aJy00Bpbvnc74Pk/view?usp=sharing

Batch 1 contains the features extracted from the 10 videos of the previous labs. Batch 2 contains the features of other 16 videos.

The data is already preprocessed, no need to apply any smoothing filters. Use the features as-is even when success is equal to zero. Linear interpolation / extrapolation was applied to deal with the unsuccessful frames.

The csv files contain the 6 action units. Note that some scenes were removed from the videos, and therefore the frames in csv files are **not always successive**. Hence, I suggest that you keep track of the frame number, to know when a scene ends, and another one begins. **You should not** end up with sequences containing two different scenes data.

After delimiting your scenes: for each scene, take each 100 successive frames alone, and if you end up with a number of successive frames less than 100, you need **to pad them** to make the size equal to 100.

Make sure to add your PAD token to your Vocab (your dictionaries): its very important to place your PAD token as your first element in the dictionary.

For each CSV file, you'll find a corresponding .f0.txt file that contains the F0 values. In each row, the first value corresponds to the "timestamp", and the second value is the F0 value.

Note that the frame rate in both files is not the same. In addition, you need to only use the F0 values that correspond to the frames in CSV (by looking at the timestamp of each frame).

The goal is for you to continue developing your seq2seq model to predict the 6 action units based on **only F0,** for simplicity.

Bonus: You can add the other features that you already extracted if you think your model is good enough with the F0 feature, and the other features are not noisy

Plotting Results

For each Action Unit, generate 3 plots to illustrate a predicted sequence Vs the ground truth sequence (for each plot).