Detecting Anomalies in Tweets using Variational Autoencoder with Reccurent Neural Networks and Inverse Autoregressive Flows

Mikkel Grønning (s144968), Søren Mørk Hartmann (s164182), Toke Bøgelund-Andersen (s164202)

1 DTU Compute, Technical University of Denmark

Acknowledgements

We wish to thank Jakob Drachmann Havtorn, Peter Ebert and the rest of Corti for the project idea and support during the project phase. A further thank to the Deep Learning Team at DTU.

Introduction

Anomaly detection in text/speech can have vital impact for in situations where humans have to make quick decisions as in emergency dispatching. If the dispatcher is assisted by AI in the shape of a trained deep neural network, it can enhance the decision making of the dispatcher in determining whether a call could come from a prankster or if the caller is experiencing unknown symptoms.

In this project anomaly detection will be carried out on tweets regarding Covid-19 where reconstruction techniques will be explored; specifically a recurrent auto encoder (RAE), recurrent variational auto encoder (RVAE) as well as recurrent variational auto encode using inverse autoregressive flows (IAF) as posterior. Based on the developed models out-of-distribution will be investigated by passing tweets from a different subject through the trained models to symbolise anomalies.

Data and Representation

As data *tweets* are used. More particular tweets regarding Corona [1] were chosen where a subset of 49,000 tweets (random subset of tweets from March 16th were selected. These are the tweets that models are trained on. For the tweets representing *out-of-distribution* tweets made by Donald Trump from 2017-2019 will be used, that can be found at [2].

In order to represent the tweets, which are text, as data that the models can interpret, two approaches have been considered; one based on characters and one based on words. The character based approach is illustrated in figure 1 where each character is mapped to an ten-dimensional space. The word based embedding is shown in figure 2 and uses a pretrained word-embedding that maps each word to a 300-dimensional space. As the word-embedding is pre-trained there are words it does not known. These words are mapped to the average of all words in the vocabulary.

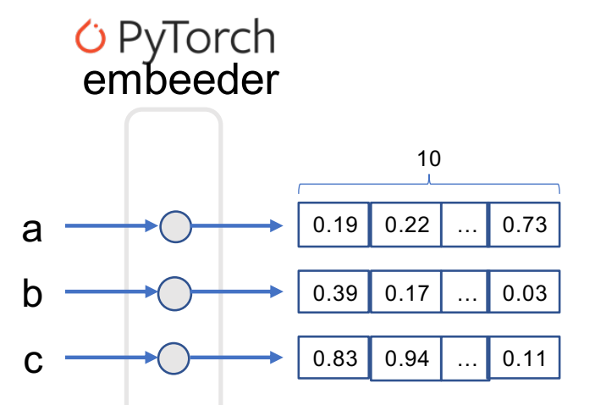

Figure 1: How characters are embedded using the Pytorch embedding function.

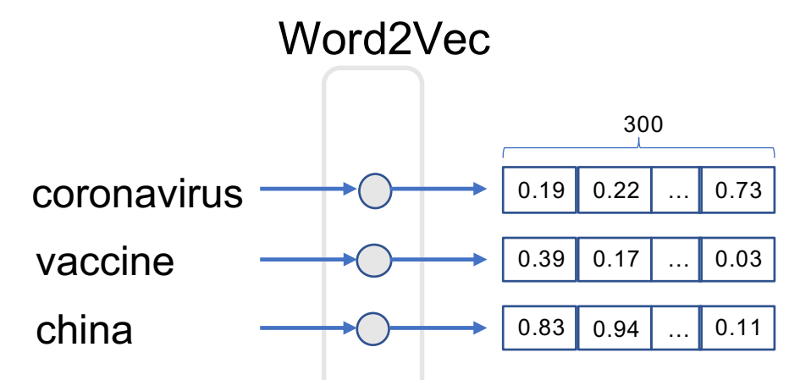

Figure 2: How words are embedded using a pre-trained Word2Vec model.

Models

Three different networks have been implemented for this projected and they are as follows.

1. An autoencoder based on a recurrent neural network in the shape of a LSTM.
2. A variational autoencoder based on a recurrent neural network in the shape of a LSTM.
3. A variational autoencoder with inverse autoregresive flow as posterior.

The three implemented models share much of the same architecture, which can be seen and compared figure 3. For all models it can be seen that LSTM is used as encoder and decoder. In addition to these three models a Latent Dirichlet Allocation (LDA) has also been trained as a benchmark.

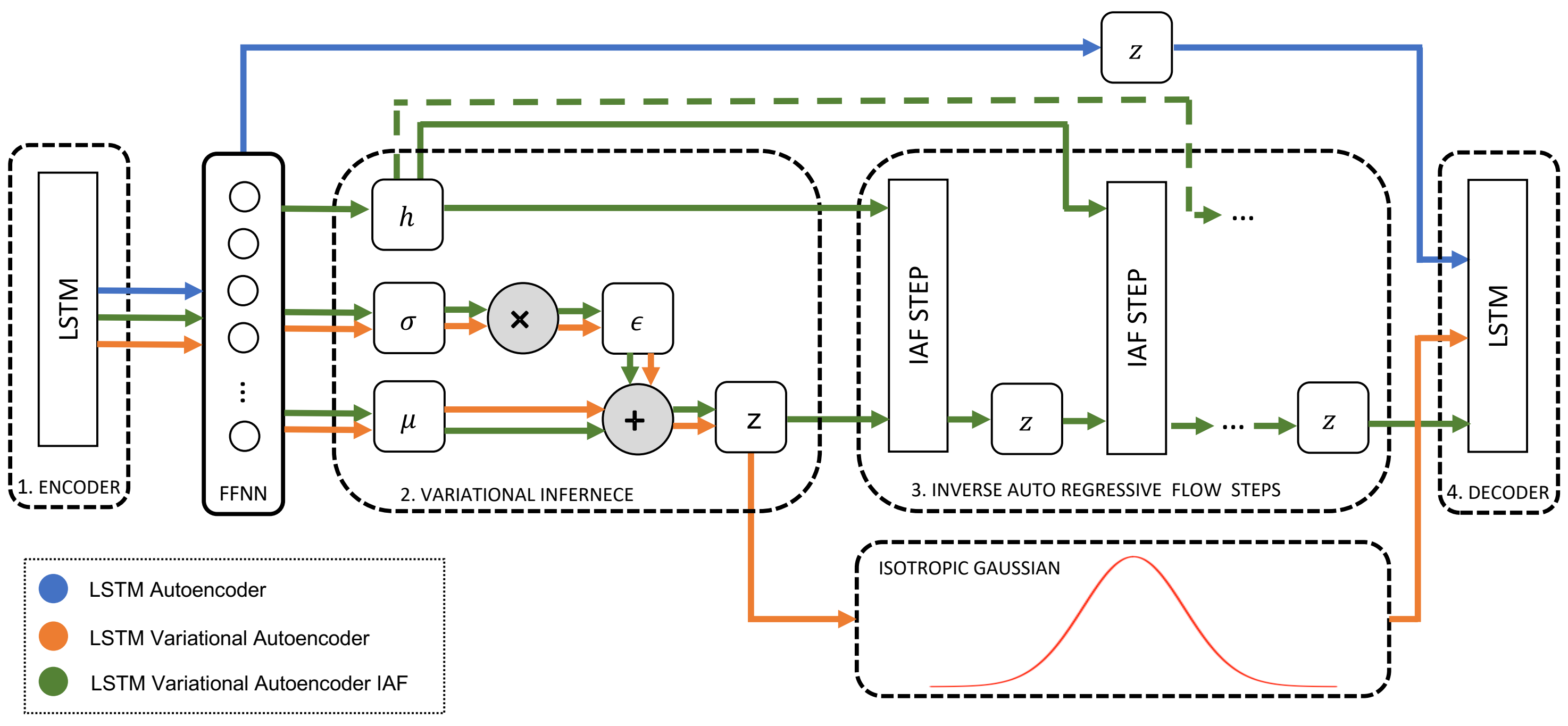

Figure 3: Model architecture for the three models used.

Theory: Inverse Autoregressive Flow

Inverse Autoregressive Flow (IAF) as posterior distribution is more powerful than the basic isotropic Gaussian. It is computationally efficient, easy to sample and in addition IAF allows for multi-modal distributions and complex relationship between the variables [3].

Inverse Autoregressive Transformations

In order to calculate the probability density for the transformed variable, it is useful for the Jacobian of the transformation to have an efficiently calculated determinant. An example of such a transformation is using an autoregressive transformation. Given a state vector $z = \{z_i\}_{i=0}^{D-1}$, we may update z' to z

$$z'_0 = \boldsymbol{\mu}_0 + \boldsymbol{\sigma}_0 \cdot z_0 \tag{1}$$

$$z'_i = \boldsymbol{\mu}_i \left(z'_{0:i-1}\right) + \boldsymbol{\sigma}_i \left(z'_{0:i-1}\right) \cdot z_i \quad i = 1 \dots D-1 \tag{2}$$

The above takes $\mathcal{O}(D)$ operations as it has to be calculated sequential. However the inverse transformation from z' to z, can be parallelized:

$$z_0 = \frac{z'_0 - \boldsymbol{\mu}_0}{\boldsymbol{\sigma}_0}, \quad z_i = \frac{z'_i - \boldsymbol{\mu}_i\left(z'_{0:i-1}\right)}{\boldsymbol{\sigma}_i\left(z'_{0:i-1}\right)} \quad i = 1 \dots D-1 \tag{3}$$

The determinant of the Jacobian of this transformation equals the product of the diagonal terms as the transformation becomes a lower triangular matrix due to the autoregressive property.

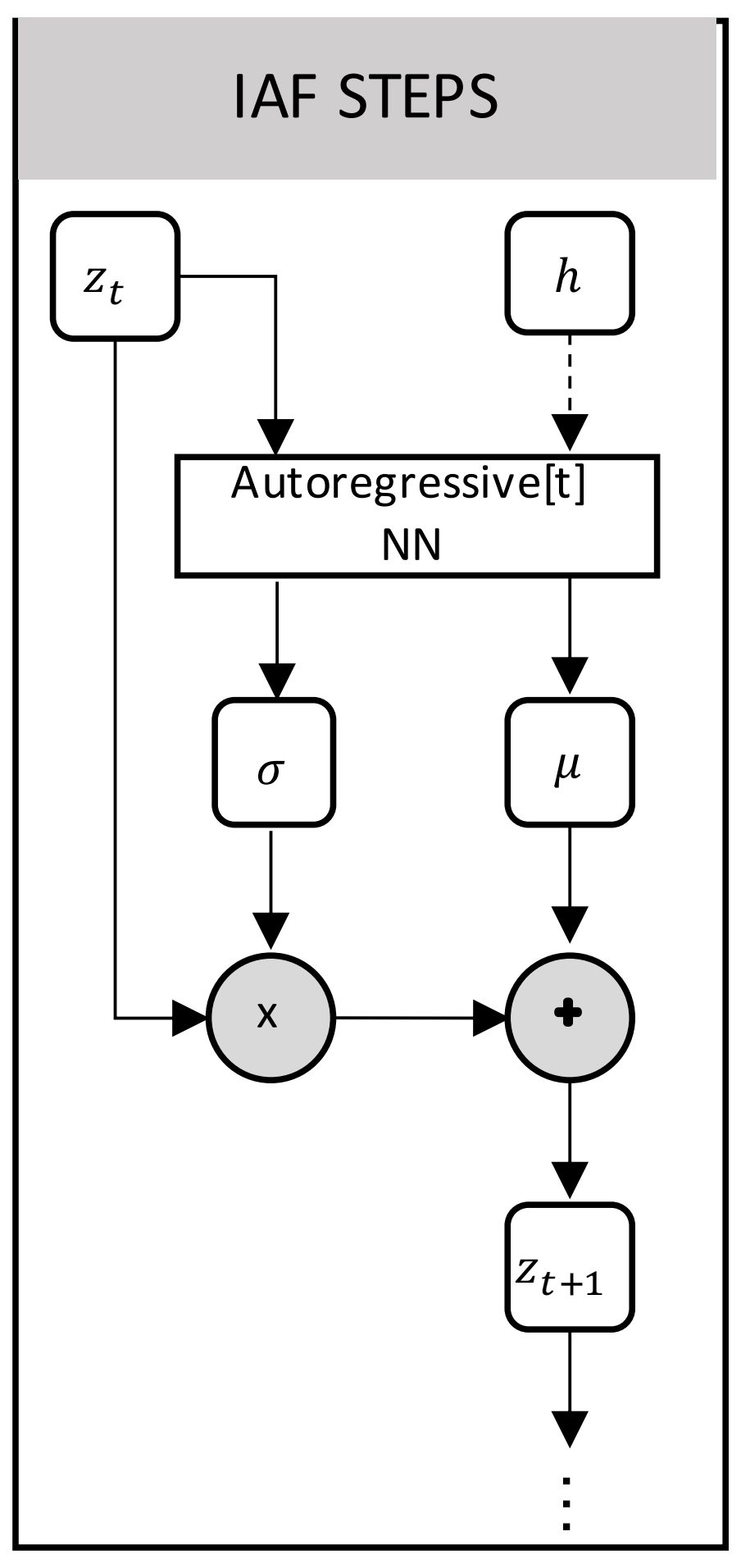

Figure 4: A single IAF transformation step.

Results

Neural Networks on toy data

The functionality of the networks are illustrated on *toy data* consisting of sequences of 2-16 identical characters with noise −1 where added as end of string element. The models are trained on 5000 observations and tested on 1000 observation . The latent space of the observations is seen in left column of figure 9. In the central column 4 points are shows and in the right-most column their reconstructions compared with the original are seen.

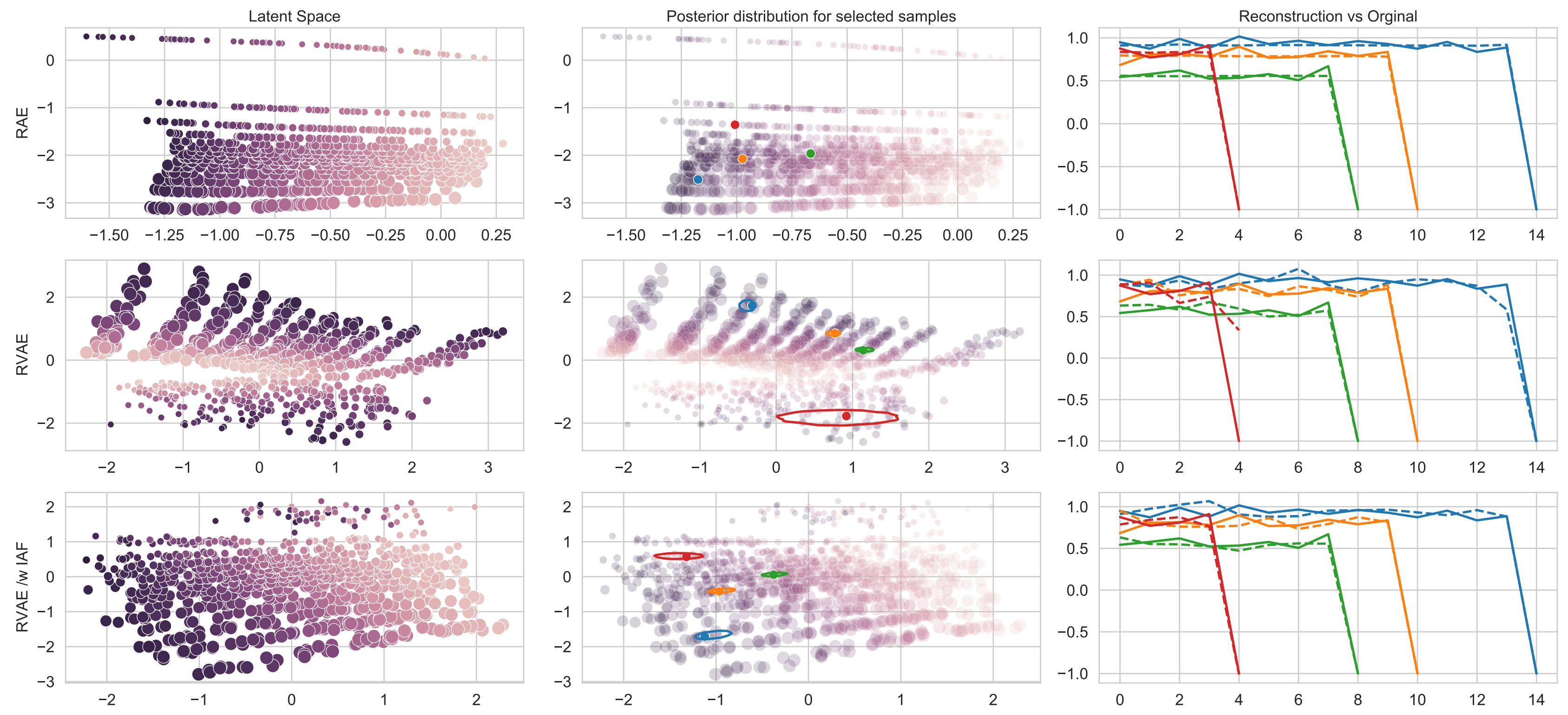

Figure 5: Overview of 1000 test-observations' placement in latent space for the three models. The scatters sizes symbolises the observations length while the color scale represents different numbers.

LDA model The LDA model was trained and Corona test data and Trump data were passed though the model. Histograms of the likelihoods are shown in figure 6, which shows some separation but predictions of new tweets would be very uncertain.

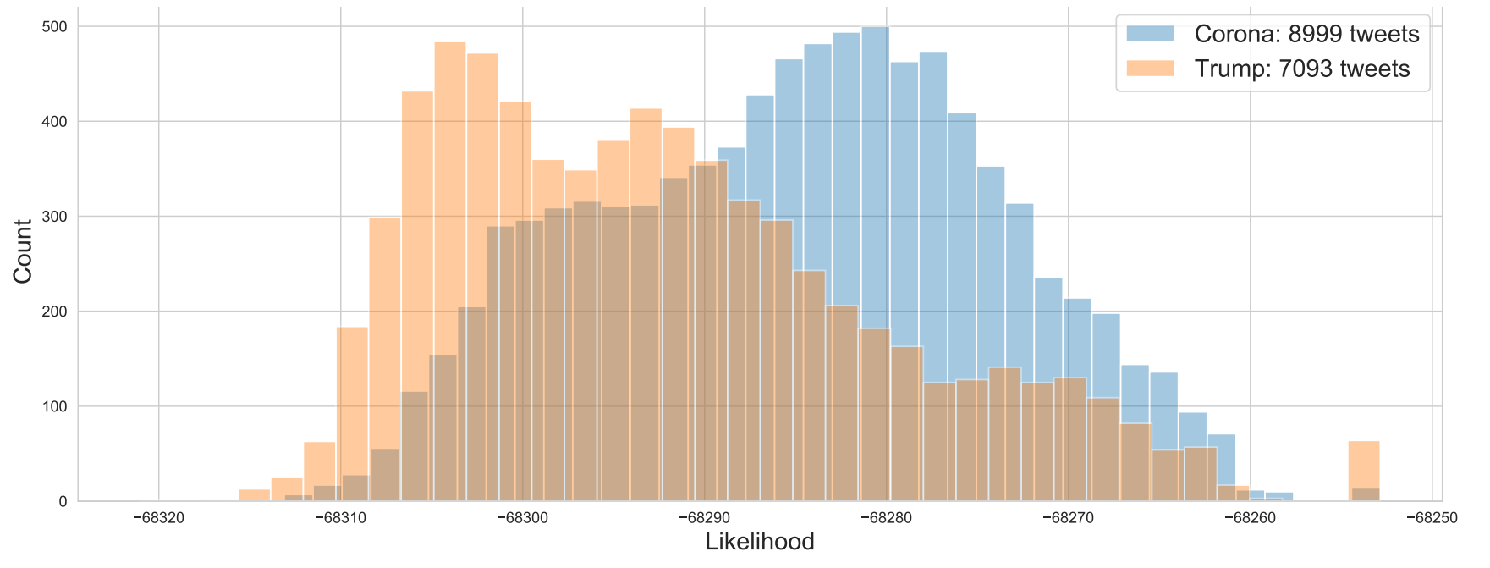

Figure 6: Histogram of likelihoods from LDA model.

Neural Networks on twitter data We applied the discussed models to the Twitter dataset. The training and validation loss can be seen in Figure 7. The RAE trained reasonably while there is still work to be done in training the variational models. Hyper-parameter tuning might help this.

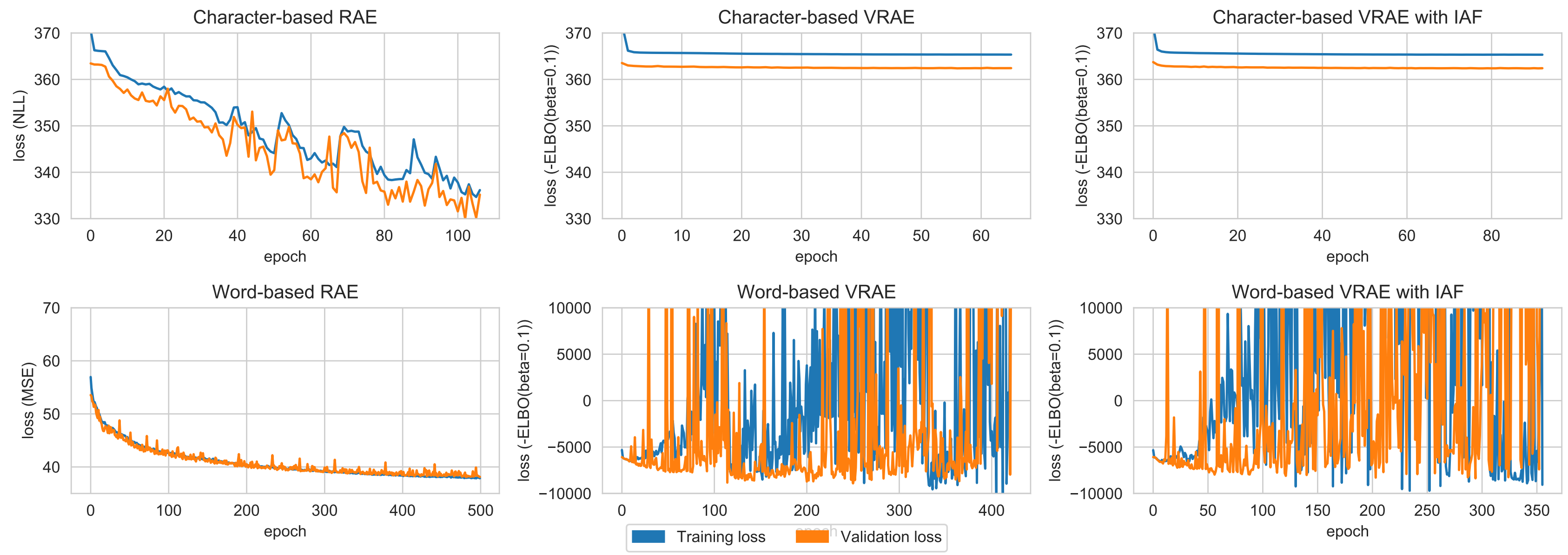

Figure 7: Convergence of models trained on twitter data

The reconstruction for RAE and log-likelihood for RVAE and VRAE with IAF for *test set* containing both Trump Tweets as well as tweets about Covid-19 are seen in Figure 8.

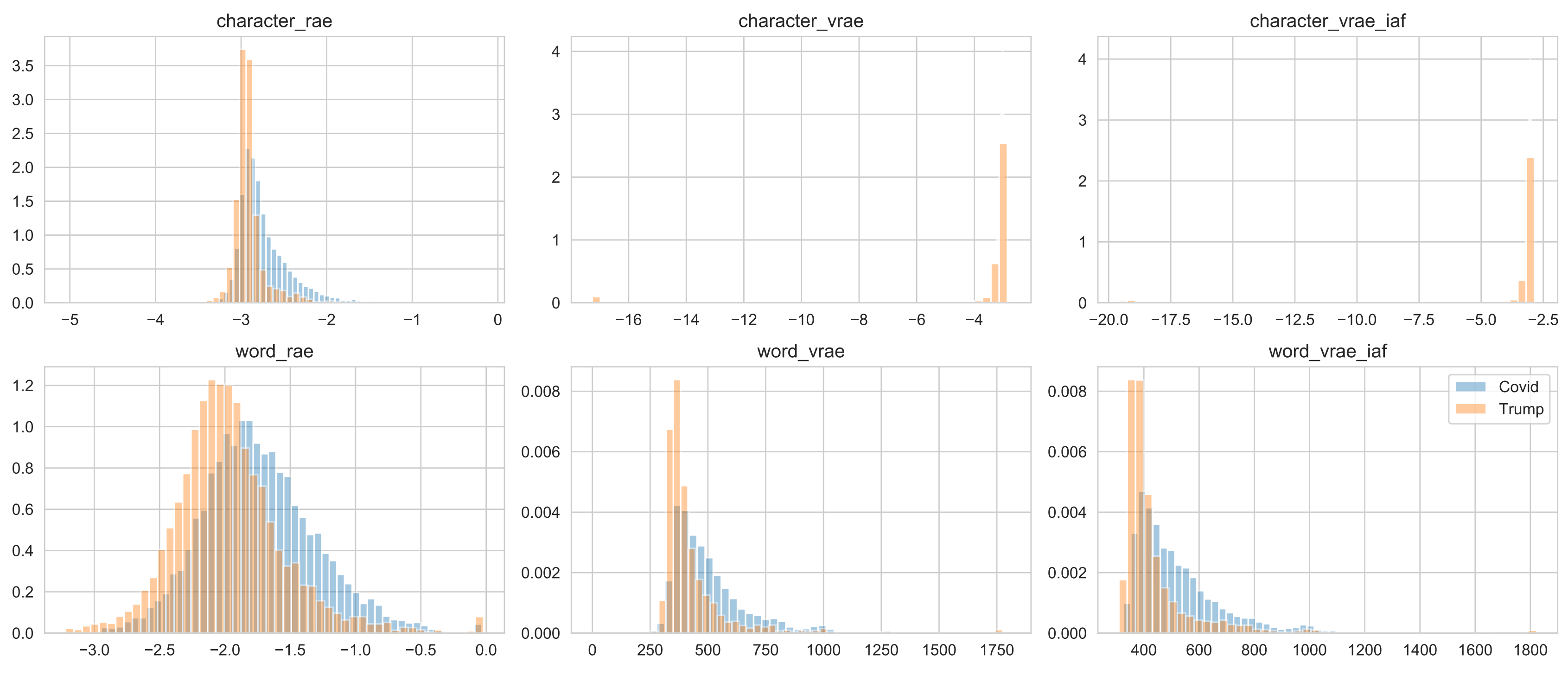

Figure 8: Histogram of likelihoods from the neural networks.

For the fun of it reconstruction of tweets using the character based model can be seen in Figure 9.

```
   RAE:  her                                                                               coronavirus    a    ...
  VRAE:  dier   siatacah   hoo r ymefpet   rovd  .a'nnn  wd1oo veyli.eu  lc snr  hbcrttosg#  unp iy  ntemaswi  mtntoe  ...
   IAF:  mastmriedigv:rhctnc'un   ci rfssb oranruiioof  ave wcciir  tet,i  oc.nboprths0thieytvnh    smnsnounu  Esshmf...
TARGET:  hilary  duff calls out Uyoung millennial  aUholes'  for Ugoing out and partying'  amid coronavirus  outbr...

   RAE:  aot  te                                                        teEEE
  VRAE:  cmtpetp  Ee ,#nacchhsu   ii iitcoontroe  aimi  sdyrr  aoah  cuereo  rxi  oE
   IAF:  vha  ioeuy   eelp    enrvebeeleaot    t g rot ingcualuyec  oeo lmEso  isno
TARGET:  and when it does end, just know this had nothing to do with it lolE

   RAE:  soc                    ccoronavirus  cEE
  VRAE:  caiah aahtlh pencoeat  ealmrotlw  oem
   IAF:  tofyfetsi'ad  rehtitstiocrtemltcpd  i
TARGET:  message #covidU19uk  #coronavirusukE
```

Figure 9: Reconstruction of tweets using characters based representation.

References

[1] J. M. Banda, R. Tekumalla, G. Wang, J. Yu, T. Liu, Y. Ding, K. Artemova, E. Tutubalina, and G. Chowell. A large-scale COVID-19 Twitter chatter dataset for open scientific research, May 2020. URL https://doi.org/10.5281/zenodo.3723939.

[2] Brendan. The trump archive, May 2020. URL https://www.thetrumparchive.com. Trump's historical tweets that can be extracted with the Twitter API.

[3] D. P. Kingma, T. Salimans, R. Jozefowicz, X. Chen, I. Sutskever, and M. Welling. Improved variational inference with inverse autoregressive flow. 2016.

[4] E. Nalisnick, A. Matsukawa, Y. W. Teh, D. Gorur, and B. Lakshminarayanan. Do deep generative models know what they don't know?, 2018.

[5] R. Vink. Distribution Estimation with Masked Autoencoders, 2019. URL www.ritchievink.com/blog/2019/10/25/distribution-estimation-with-masked-autoencoders.

[6] R. Vink. Another normalizing flow: Inverse Autoregressive Flows, 2019. URL https://www.ritchievink.com/blog/2019/11/12/another-normalizing-flow-inverse-autoregressive-flows/.