CHATBOT 分享

赵英俊

HTTPS://GITHUB.COM/ZHAOYINGJUN

· chatbot项目工程简介

- 整体工程架构
- 配置文件和web工程
- 适配的框架版本

· SeqGAN版本分享

- SeqGAN原理简介
- SeqGAN实现过程设计
- 核心代码分析

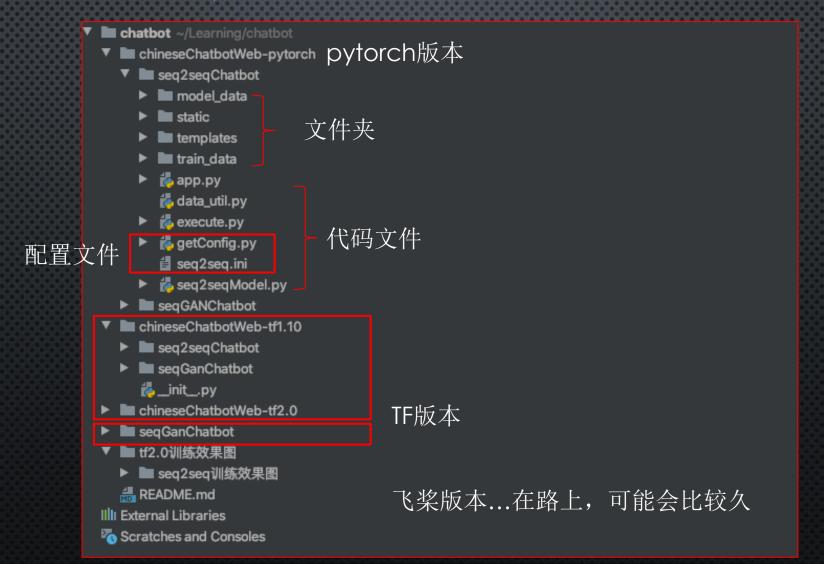
• 核心模块介绍TF2.0 vs pytorch

- seq2seqModel
- execute
- app

• 项目路标

- SeqGAN的TF2.0 pytorch版本更新
- 飞桨平台适配
- 多轮情感聊天机器人版本

chatbot项目工程简介



核心模块介绍 Model=encoder+attention+decpder+loss

▼ lachineseChatbotWeb-pytorch ▼ laseq2seqChatbot ▶ model_data ▶ **static** ► **templates** ► train_data # app.py 🖧 data_util.py & execute.py getConfig.py # seq2seq.ini # seq2seqModel.py AttentionDencoder(nn.Module) © Encoder(nn.Module) f train_step(input_tensor, target_tensor, encoder, decoder, encoder_optimizer, decoder_optimizer) ► seqGANChatbot ► chineseChatbotWeb-tf1.10 ▼ lack chineseChatbotWeb-tf2.0 ▼ lim seq2seqChatbot model_data ▶ **static** ► templates ▶ **train_data** 🖧 data_util.py ▶ ♣ getConfig.py # seq2seq.ini seq2seqModel.pv BahdanauAttention(tf.keras.Model) © Decoder(tf.keras.Model) © Encoder(tf.keras.Model) f loss_function(real, pred) f train_step(inp, targ, targ_lang, enc_hidden)

核心模块介绍

execute=create_data+preprocess_sentence+tokenize+train+predict

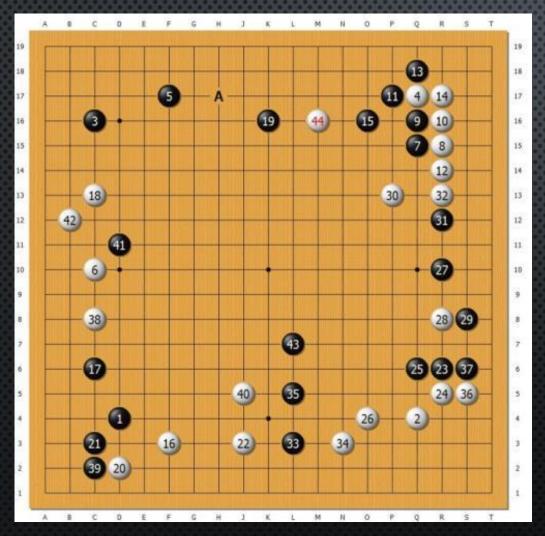
▼ laseg2segChatbot ▶ model data ► **static** ▶ templates ► train_data data_util.py execute.pv f create_dataset(path, num_examples) findexesFromSentence(lang, sentence) ▶ G Lang max_length(tensor) predict(sentence) f preprocess_sentence(w) f read_data(path, num_examples) tensorFromSentence(lang, sentence) f train() seq2seq.ini ▶ **seqGANChatbot** ► chineseChatbotWeb-tf1.10 ▼ lachineseChatbotWeb-tf2.0 ▼ laseq2seqChatbot model_data ▶ ■ static ▶ templates ► **train_data** app.py data_util.py create_dataset(path, num_examples) max_length(tensor) predict(sentence) f preprocess_sentence(w) fread_data(path, num_examples) f tokenize(lang)

f train()

核心模块介绍 app=heartbeat+index+reply

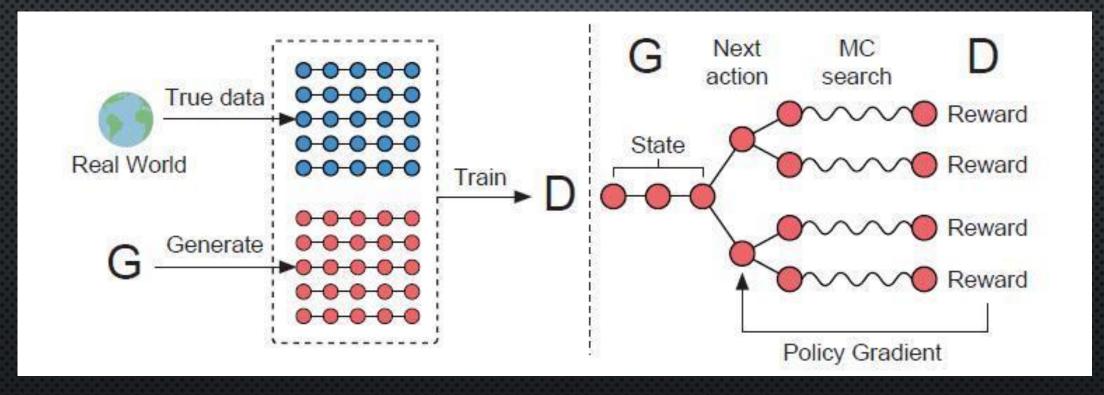
```
▼ ■ seq2seqChatbot
    ▶ model_data
    ▶ ■ static
    ▶ templates
    ▶ train_data
    ▼ 🐉 app.py
        f heartbeat()
        f index()
        f reply()
      👸 data_util.py
    detConfig.py
      general seq2seq.ini
    ▶ seqGANChatbot
▶ matchineseChatbotWeb-tf1.10
▼ lachineseChatbotWeb-tf2.0
  ▼ laseq2seqChatbot
      model_data
    ► static
    ▶ templates
    ► train_data
    ▼ 👸 app.py
        f heartbeat()
        f index()
        f reply()
      🖧 data_util.py
    getConfig.py
      general seq2seq.ini
    k seq2seqModel.py
```

SeqGAN版本分享: 围棋和自然语言一样都是离散的,那么围棋可以,自然语言也可以。



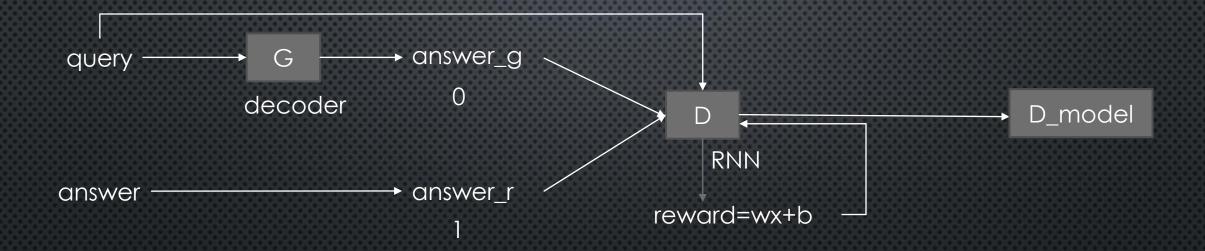
围棋和语言一样都是离散的,那么围棋可以,自然语言也可以 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20

SeqGAN版本分享: seqGAN原理(内力不行,外力驱动)

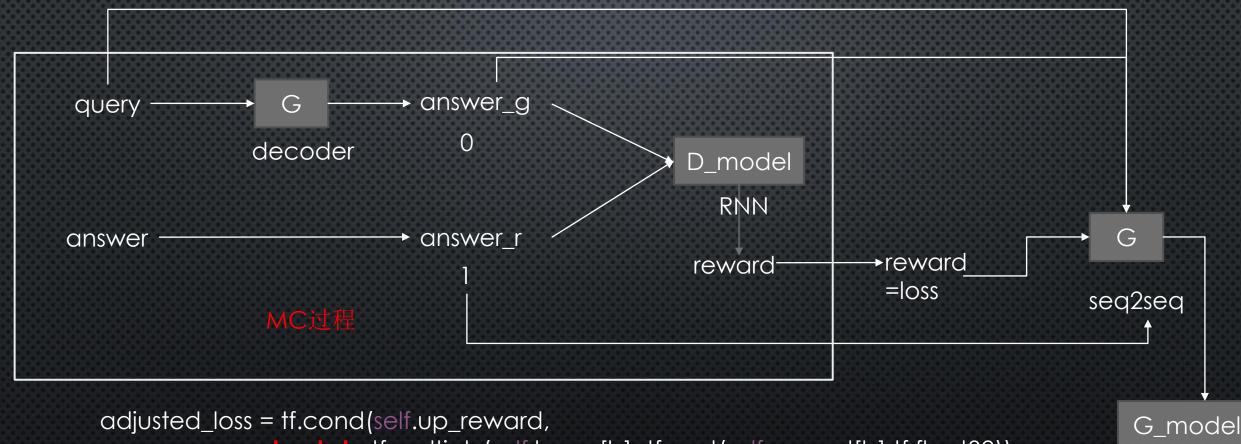


自然语言序列是离散的,那么单纯的GAN输出也是离散的,导致梯度不可导,无法驱动参数更新。 那么就用策略梯度来驱动,根据Reward来驱动逼近极值。

SeqGAN版本分享: seqGAN实现过程设计-更新D



SeqGAN版本分享: seqGAN实现过程设计-更新G



SeqGAN版本分享: seqGAN核心代码分析

```
▼ seqGanChatbot
 ▼ ladisc
     🛵 __init__.py
   discriminator.py
   rnnModel.py
 ▼ 🖿 gen
     🐔 __init__.py
   to gen_model.py
   static
 templates
 ▶ utils
   🛵 __init__.py
 app.py
   a execute.py
 🛵 __init__.py
```

SeqGAN版本分享: seqGAN实现过程设计-更新G

```
def decoder(num_roll):
    for _ in xrange(num_roll):
        _, _, output_logits = gen_model.step(sess, encoder_inputs, decoder_inputs, target_weights, bucket_id,
                                             forward_only=True)
        seq_tokens = []
        resps = []
        for seq in output_logits:
            row_token = []
            for t in seq:
                row_token.append(int(np.argmax(t, axis=0)))
            seq_tokens.append(row_token)
        seq_tokens_t = []
        for col in range(len(seq_tokens[0])):
            seq_tokens_t.append([seq_tokens[row][col] for row in range(len(seq_tokens))])
        for seq in seq_tokens_t:
            if data_utils.EOS_ID in seq:
                resps.append(seq[:seq.index(data_utils.EOS_ID)][:gen_config.buckets[bucket_id][1]])
            else:
                resps.append(seq[:gen_config.buckets[bucket_id][1]])
        for i, output in enumerate(resps):
            output = output[:answer_len] + [data_utils.PAD_ID] * (answer_len - len(output) if answer_len > len(output) else 0)
            train_query.append(train_query[i])
            train_answer.append(output)
            train_labels.append(0)
    return train_query, train_answer, train_labels
if mc_search:
    train_query, train_answer, train_labels = decoder(gen_config.beam_size)
else:
    train_query, train_answer, train_labels = decoder(1)
return train_query, train_answer, train_labels
```

SeqGAN版本分享: seqGAN核心代码分析

```
seqGanChatbot
 ▼ ladisc
      __init__.py
   discriminator.py
   frnnModel.py
 ▼ lgen
      🛵 __init__.py
   ▼ 🐉 gen_model.py
      Seq2SeqModel(object)
   generator.py
    ► 🖧 seq2seq.py
 ▶ static
 ▶ templates
 ▼ utils
      __init__.py
   conf.py
   data_utils.py
    __init__.py
 dapp.py
 ▼ 🐉 execute.py
      f al_train()
      decoder_online(sess, gen_config, model, vocab, rev_vocab, inputs)
      f disc_pre_train()
      f disc_step(sess, bucket_id, disc_model, train_query, train_answer, train_labels, forw
      f disc_train_data(sess, gen_model, vocab, source_inputs, source_outputs, encoder_ir
      f gen_disc()
      f gen_pre_train()
      f gen_test()
      finit_session(sess, gen_config)
      f main(_)
      f softmax(x)
```

- 项目路标:

- -SeqGAN的TF2.0 春节期间更新, pytorch版本更新待定, 预计在国庆节期间
- -飞桨平台适配-21年6月
- -多轮情感聊天机器人版本-待定