Katholieke
Universiteit
Leuven

**Department of
Computer Science**

CONVERSATIONAL QUESTION ANSWERING
A Generative Modelling Approach

Siddharth Agarwal (r0773458)

Academic year 2019–2020

Contents

List of Figures

Abstract

Conversational Question Answering (CoQA) is a dataset released by Stanford NLP. It consists of conversations and questions asked on the basis of those conversations. Answering these questions is a trivial task for humans but one not nearly so for learning models. We have known seq2seq models that have performed remarkably well on other QA datasets such as SQuAD, but most of those fail to match that performance on CoQA. Furthermore, reading comprehension models that have performed remarkably on standard NLU tasks have failed to perform well on this task as well. In this paper, I propose two deep generative models for the task of multi-turn question answering that leverages the capability of generative Adversarial networks and variational autoencoders to learn data distributions to develop richer output spaces.

1 Introduction

Natural language processing is broadly divided into two categories based on task definition: natural language generation (NLG) and natural language understanding (NLU). These tasks, while not necessarily separate, do require different modelling approaches. NLG is a task that has historically leaned more heavily on understanding local context and providing plausible outputs. NLU is more focused on retrieving relevant data from the corpus, obtaining insights from the text data and providing information related to it. Tasks like named entity recognition, semantic role labelling, and machine comprehension broadly fall into this category.

However, it makes sense to think of these tasks as sub-components of a larger NLP task-space. If we draw an analogy with human intelligence, we perform the tasks of understanding and generating language together. This is usually seen in tasks such as machine translation, story completion, and question answering. All of these tasks require understanding non-local context and generating outputs that are coherent, plausible, and satisfy the semantic constraints imposed by the prior context, i.e. a translated sentence should have the same meaning, a generated answer should actually answer the question, etc. These tasks have been approached in a deep learning setting via discriminative methods such as LSTMs, seq2seq models, or transformers. These, in combination with distributed language models, have proven massively successful on these tasks compared to previous methods applied to them.

In this paper, I attempt to move past the discriminative modelling regime, and suggest that deep generative modelling approaches, that have been underexplored in the context of question answering, can be used to achieve strong results on the task.

The paper will first detail a history of question answering to provide context for the methods that have already been tried on the task to give the reader a better appreciation for the successes achieved in it. We will then look at the Conversational Question Answering dataset, and the methods that have achieved success with it. I will then motivate the deep generative models we plan to use. After that, I will describe the models and provide a discussion for them.

2 A Brief History of Question Answering

Question Answering (QA) has long been a benchmark problem in NLP. It is formally defined as a sophisticated form of information retrieval characterised by information needs that are at least partially expressed as natural language statements or questions, and it is one of the most natural forms of human computer interaction.[15] It has long been of interest from an information retrieval perspective in terms of application. It has also been of interest from a Natural Language Understanding (NLU) perspective in terms of fundamental advances in the domain.

There have been attempts to work on the problem practically since the notion of Artificial Intelligence (AI) has existed. Some of the first attempts were the Baseball, LUNAR, and SHRDLU models from 1960's and early 70's that were designed to answer questions or carry dialog about restricted domains posed in natural language. These models were heavily symbolic and relied on dictionaries to "read" words and idioms and then use phrase-structure grammars. This basically transformed a natural language question into a database query, and then following some content analysis, generated an answer from a knowledge base. [10] These systems were brittle, limited in scope, and relied very heavily on expert-crafted knowledge bases.

The 1980's saw the advent of expert systems and in turn QA systems that were based on these. These systems harnessed the rapid development in computational linguistics that was achieved in that decade to comprehend questions asked in natural language and provide answers. One such system was the Unix Consultant [28]. This again relied on expert-crafted knowledge bases. These systems again used relatively sophisticated linguistic techniques to represent natural language questions as logic statements and in turn used these statements as database queries. They were again brittle and did not "understand" natural language quite well.

These systems, while successful in very limited domains, suffered from a lot of common flaws. They really could not represent natural language but could only apply grammars to them and perform some entailment. As an example, the UC model requires questions to be prefixed with a #. This meant that unless questions were phrased in a very particular way, they were likely to be misinterpreted by the system. Another major flaw was the heavy reliance on hand-crafted knowledge bases. These would quickly get cumbersome to maintain and would have a lot of difficulty scaling to large domains.

In the late 1990's, open-domain question answering, i.e. answering questions that require the model to query large unstructured text corpora, was instantiated by TREC [24]. This was released as a dataset on which an annual competition was held. This was a small-scale dataset compared to modern datasets such as SQuAD. It had 200 questions and a series of documents which were confirmed to have the answers. A problem that was noticed while this dataset was developed was that it was difficult for multiple reviewers to agree on what an "objectively good" answer to a question is. This competition has now been held every year since 1999 and owing to the ever-increasing computational capacity, the size of the dataset has continued to increase. This dataset, however, is not entirely open-domain. It is interesting in the sense that it does not separate questions that are open-domain or restricted.

Here I expand upon the important distinctions between open-domain and restricted-domain question answering. This will have an effect on the targets of a model and how effective it will be:

- Open-domain Question Answering:
  - Larger datasets can be constructed as topics in consideration are as varied as human conversations can be.
  - General sources such as wikipedia can be used as a knowledge base for these datasets, and regular people can be used to create the dataset.
  - However, because of the vaster range of topics, the models have trouble generalising, and therefore answers are of a less-than-stellar quality.
- Restricted-domain Question Answering:
  - Smaller datasets as there is, by definition, a limited domain to work in.
  - They require domain experts to construct the datasets, and specialised knowledge bases are required.
  - Models trained in this domain provide answers of a higher quality.

In the 2000's there were many techniques that received focus in the question-answering domain. Because of the advent of multimedia data (I am sticking to text in this paper) becoming more commonplace and the necessity of retrieving information from the internet, QA was treated as an information retrieval problem. Another important development during this time were more sophisticated phrase-structure grammars that allowed syntax and semantics to be represented together. This, along with the logical knowledge representation paradigms formed the basis for the question answering systems of the 2000's. Techniques such as semantic role labelling, textual entailment, and defining the type of answer expected saw a surge in usage [15]. At the same time, the complexity of datasets continued to increase, as did the complexity of the problems posed in the context of question-answering. As ane example, multilingual question-answering was presented as a problem. All models in this era consisted of rather heavy feature engineering, that often contributed to model complexity and required domain-specific knowledge.

In the previous decade, with the advent of deep learning and distributed semantics, question answering saw massive improvements, sometimes with much simpler models than those that preceded them. As an example, Yu et al. [30] showed that models developed using deep learning and no feature engineering or linguistic tools could provide SOTA results on datasets such as TREC. This model performed answer sentence extraction, which in a strict sense, might not answer just the question but provides the sentence in which the answer is contained. In this case, the problem was represented as a binary classification model with the categories being a correct or a wrong answer. More sophisticated neural network architectures as convnets and LSTMs were used to represent data and generate answers for questions.

Another family of models that use deep learning and have shown promise come from the notion of neural semantic matching. One of the most interesting methods that use this is the ANMM (Attention-Based Neural Matching Model) [29] which uses an attention mechanism focused on the value-based weighting instead of position-based weighting that is the standard in attention models. It is a seq2seq model [21] similar to those that saw great success in machine translation. It achieved near SOTA results on TREC that year, while again being an extremely simple model.

A logical conclusion to these models would be something that combines both methods described above. It can use a representation-based model to generate question and answer embeddings, and use a matching scheme to generate a score using which the answer quality can be tested. A model that uses Bi-LSTMs to generate representations and multiple positional matching scores to get the answer-sentence that fits best was made by Wan et. al [25].

More modern methods that work on QA tasks have been developed using transformers, knowledge modelling, and pretrained contextual word embeddings. These methods have outperformed humans on the metrics in many cases.

3 The CoQA dataset

The Conversational Question Answering dataset [20] is a dataset meant to bridge the space between dialogue systems, reading comprehension, and question answering systems. It makes sense to have such a dataset, as humans do not really differentiate between these tasks. To us, they are all usually a part of normal conversation. Beyond this, a dialogue system will have to have the capacity to answer questions from prior context. To encourage robustness in systems that can handle these tasks, the dataset was developed. It contains 127000 questions taken from 8000 conversations from seven different domains. It has unanswerable questions, in some places. The dataset is structured as having a passage and then a series of questions with their answers.

Another visible way in which the dataset departs from existing QA datasets is the focus on "naturalness" in both the questions and the answers. Some questions are contextual, a single word following from previous questions and answers, like "Who?" based on a description provided previously. Similarly, answers are, in places, brief and contextual. Most standard QA dataset answers are long contiguous spans of text which, while useful, is distinct from the way conversation usually flows amongst humans. The answers are also not necessarily structured as proper text. 33.2% of answers in the dataset are free-form.

Another aspect of the dataset that has to be considered is the coreference chain it has integrated. Entities are often mentioned once and then referred to using pronouns and common nouns. This adds an additional entailment challenge.

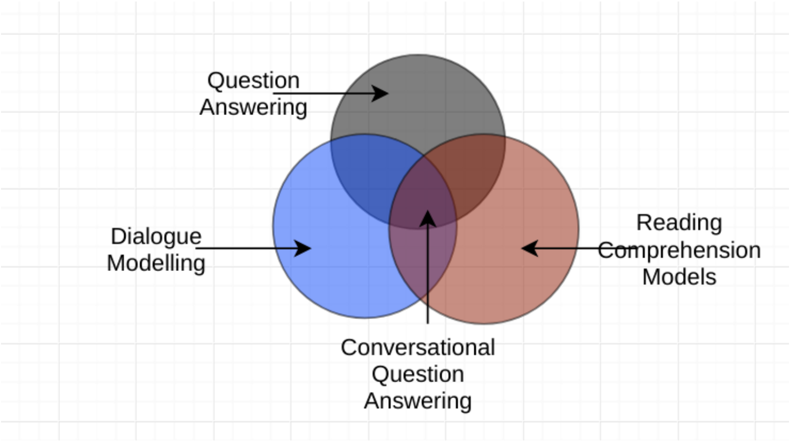
Question Answering
Dialogue Modelling
Reading Comprehension Models
Conversational Question Answering

Figure 1: Intuitive visualisation of the task definition of CoQA

3.1 Models for CoQA

In this subsection, I will describe two of the models that have been effective on the CoQA to provide a notion of the current state of the art. Most of the model in this space have relied on variants of BERT [7].

The model [13] currently at the top of the CoQA leaderboard uses a combination of the following components:

- RoBERTA, a BERT pre-training approach to contextual language modelling.
- Knowledge distillation [6], a technique wherein a model (the teacher)'s output is the objective for another model (the student). This leads to the student outperforming the teacher even if they have the same architecture.
- Adversarial training, as defined by Goodfellow et al. [9] which helps it generalise better.
- Rationale tagging multitask, which makes the model choose whether each token of the paragraph should be included in the rationale.

The model performs quite well on the dataset as evidenced by it being at top of the leaderboard, but falters on the free-form answers, a problem acknowledged in the paper itself.

Another model that performs quite well on CoQA is the one developed by Google for SQuAD 2.0 [19], another major QA dataset. This dataset was a single turn dataset unlike CoQA, and did not require the model to keep track of previous states.

The base model, as detailed in Ohsugi et al. [18], relies on BERT and takes as its input, the input and the output tokens, and learns a representation for them that can generate a relevant output given the input. This is worth noting, as the model does not necessarily learn the distribution. It learns the discriminative relationship between the questions and the answers.

The advance that the authors proposed to adapt the model for CoQA is independently conditioning the context paragraph with each question and answer, providing a multi-turn context.

4 Deep Generative Modelling

Deep generative modelling became one of the most interesting topics in deep learning research with the advent of GANs and VAEs. I will be considering both of these in the ideas presented and the models I am proposing. We describe them below:

4.1 GANs

These models can be defined as an adversarial process, in which we simultaneously train two models: a generator G that captures the data distribution, and a discriminator D that estimates the probability that a sample came from the training data rather than G. The training procedure for G is to maximize the probability of D making a mistake. This framework corresponds to a minimax two-player game.[9] They have achieved a high level of success in generating realistic data, and variants of it have been quite successful in tasks such as style transfer [14], conditional data generation [11], text-to-image translation [23] amongst others.

I present the GAN loss and the architecture, but a detailed discussion of GAN training would be outside the scope of this paper.

The GAN loss is something worth looking at. It can be optimised with backpropagation, which makes it rather unlike the generative models such as Bayesian Networks and HMMs. Most of them relied on approximate inference or Markov chains. The standard GAN loss looks as shown below:

$$min_G max_D V(D,G) = \underset{x \sim p_{data}}{\mathbb{E}}[log(D(x))] + \underset{z \sim p_z(z)}{\mathbb{E}}[log(1 - D(G(z)))] \tag{1}$$

where $x \sim p_{data}$ indicates that x is from the input data distribution, and $z \sim p_z(z)$ indicates that z is from the random noise distribution.

The architecture for the basic GAN is described in Figure 2

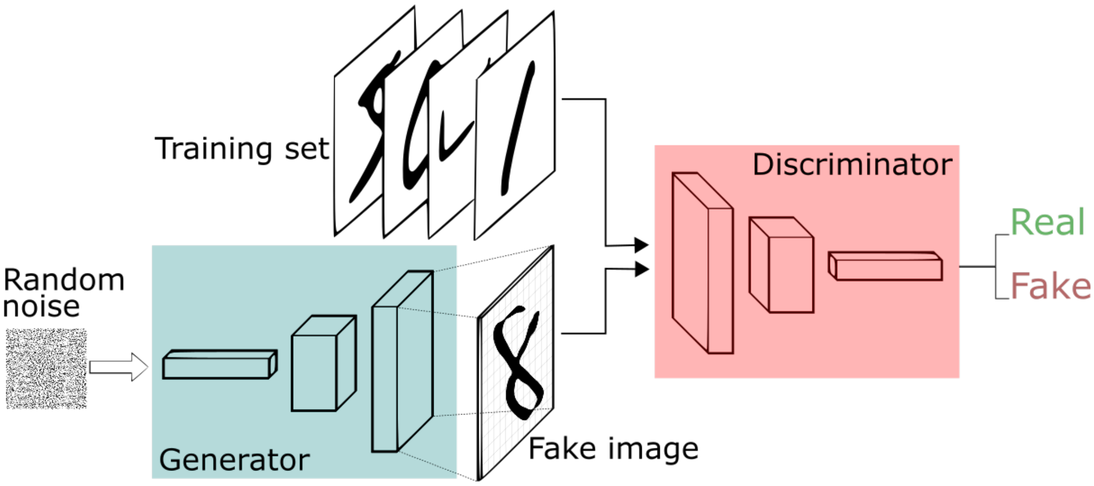

Figure 2: An illustration of a Generative Adversarial Network Architecture [1]

4.2 Variational Autoencoders

Variational Autoencoders (VAEs) are a recent advance in generative modelling of data. They are based on the autoencoder [4]. They share a similar architecture, an encoder and a decoder that outputs data of the same dimensionality as the input.

A VAE is best thought of as an autoencoder the training of which is regularised such that the latent space, i.e. the representation from the encoder that is passed to the decoder, has properties that allow the generative process. This latent space can be constrained to adhere to a distribution, the variance of which can be used to modify the inputs. The loss function for the basic VAE relies on minimising the Kullback-Liebler divergence between the input and the latent space.

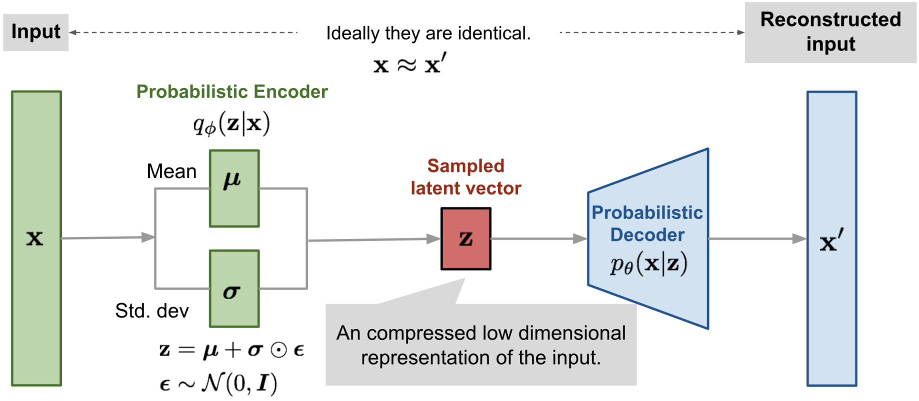
Input
Ideally they are identical.
$\mathbf{x} \approx \mathbf{x}'$
Reconstructed input
Probabilistic Encoder
$q_\phi(\mathbf{z}|\mathbf{x})$
Mean
$\boldsymbol{\mu}$
$\mathbf{x}$
$\boldsymbol{\sigma}$
Std. dev
Sampled latent vector
$\mathbf{z}$
Probabilistic Decoder
$p_\theta(\mathbf{x}|\mathbf{z})$
$\mathbf{x}'$
$\mathbf{z} = \boldsymbol{\mu} + \boldsymbol{\sigma} \odot \boldsymbol{\epsilon}$
$\boldsymbol{\epsilon} \sim \mathcal{N}(0, \boldsymbol{I})$
An compressed low dimensional representation of the input.

Figure 3: Illustration of variational autoencoder model with the multivariate Gaussian assumption. [2]

5 Motivation for deep generative model for CoQA

5.1 Weaknesses of current NLG models

As we can see in Section 3, any model that seeks to be effective on the CoQA dataset will need to have a generation component. I will not belabour the details of NLG here but will list the drawbacks of current NLG models.

Gao et al. [8] list a few weaknesses with modern dialogue models. These are generalisable to current NLG. They are listed below:

- Bland responses: This is because current NLG models are trying to maximise a discriminative probability $P(T|S)$ where T is the target corpus, and S is the source. It makes sense that the blander answers will work in a larger number of circumstances, given the source. Generative modelling techniques such as adversarial training have been used on these tasks to improve the variety of the answers developed.

- Word Repetitions: This is a common NLG problem in tasks that are not quite as one-to-one constrained as machine translation. A word in the context/source can map to multiple words in the target embeddings, leading to the model generating the same words frequently. Implementing self attention on the decoder has been presented as a potential solution to this.

5.2 Deep Generative Modelling on Natural Language Processing tasks

I have briefly introduced deep generative models in the previous sections. They have seen tremendous success in image and audio processing tasks but their usage in Natural Language Processing (NLP) tasks remains limited currently. Amongst the first problems we will encounter on this task come from the fact that text data tends to be discrete. This can be overcome by using categorical reparameterisation, possibly using the Gumbel-Softmax method presented by Jang et al [12]. Another approach can be to use convolutional LSTMs, or in more recent research, transformers to obtain smooth localised representations at the sentence and document levels. There has been some interesting research in NLU and NLG I shall talk about in order to motivate our model.

5.2.1 GAN-based dialogue generation

Amongst the first models to integrate GANs in dialogue generation systems was proposed by Li et al. [16]. This model, inspired by the Turing test, uses a GAN-like architecture wherein a generator attempts to generate dialogue while a discriminator, taking the place of a human in the Turing test setting, tries to distinguish between actual dialogue and machine-generated dialogue. The outputs of the discriminator were used as rewards for the generator, adding a RL component to the model. This was an interesting beginning as, while the authors saw improvement compared to the baselines, they did not see SOTA performance.

5.2.2 GAN-based machine comprehension

Another work that relies on GANs to solve the machine comprehension task was presented by Wang et al [26]. This model relies on conditional GANs, with the potential outcomes as the conditioning variables, and a gated recurrent unit as the architecture for both the generator and the discriminator. It uses the *Story Cloze Test* as the commonsense machine comprehension dataset and has the task of inferring commonsense entailments from text data. The model is provided with candidate answers and must select the sensible output based on the given context. The model, despite using relatively unsophisticated word2vec embeddings, achieves SOTA performance on the task because of the GAN architecture.

5.2.3 VAE-based story generation

Similar to GANs, VAEs have found success in NLP tasks. Amongst the most interesting VAE results in machine comprehension and NLG come from Wang et al [27]. The model they proposed is a transformer-based conditional VAE. The task they intended to use it on is story generation on the *ROCstories.* The model, utilising the richness of representation achieved by a VAE, was able to generate stories that offered story plots that ranked high in readability, diversity and coherence. The model was able to achieve SOTA on the task, indicating that VAEs have potential in NLP. I present the TCVAE architecture in 4

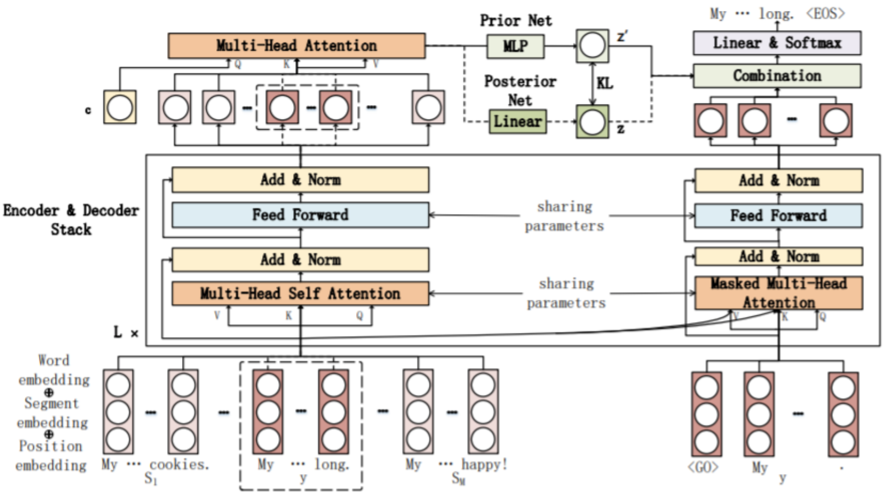

Figure 4: The TCVAE architecture [27]

5.2.4 VAE-based dialogue modelling in low-resource languages

Another interesting use-case for VAEs in NLG comes from Tran et al.'s [22] work on low-resource NLG tasks such as dialogue generation. This model uses a dual-VAE architecture. It consists of a variational generator, and a Variational CNN-DCNN model. They are trained together to perform dialogue generation, and the model generalises well despite the clear lack of data in the languages. The models achieved SOTA results even on language datasets that were extremely minimal.

6 Our Models

In the previous section, I presented work done using deep generative modelling on NLU and NLG tasks. We saw that these models perform remarkably well on multiple tasks and often achieve SOTA performance. However, none of those tasks combined NLU and NLG quite the way the task of conversational question answering does. To the best of our knowledge, there has been no work done on conversational question answering that utilises the capabilities of deep generative modelling. In this section, I will present two models, one based on the GAN, another based on the VAE, as potential techniques to improve performance on the task.

6.1 GAN-based model for CoQA

I detail the model to tackle the CoQA task using a conditional GAN-based setting here. The two-step procedure proposed in Ohsugi et al. [18] is modified to fit our task.

6.1.1 Contextual Embeddings

The first step involves developing contextual embeddings. These will be used for both the generator and the discriminator. We will fine-tune the embeddings for the discriminator as they will allow the discriminator similar context to judge output. This will allow the generator to learn a latent representation which will be more sophisticated.

In this part of the modelling process, we are trying to obtain contextual embeddings of an output sequence , given the input query(x), i.e. the question relevant to this output, and the context paragraph(y), using BERT. We are trying to model

$$\mathbf{a} = f(BERT(x, y|\theta)) \tag{2}$$

The BERT function outputs the d-dimensional BERT outputs and the f function extracts the relevant context paragraph segments.

Where the method differs from traditional pre-training is that the questions are encoded in a similar manner, and each question embedding includes a question history component. What this means is that each question embedding will have a separate embedding component that will contain information about the questions that have been asked. Similarly, there will be an answer history embedding which contains the information about the answers to each of the previous question. It is easy to see that this mode of embeddings can easily be effective in a conversational QA setting.

6.1.2 The Generator

In this section, I describe the conditional GAN generator architecture. The generator will be using the fine-tuned embeddings obtained in the previous step as the input. The target for the generator is to produce the answer text span. The output of the generator should be such that the discriminator is unable to discriminate between the generated output and the ground truth output, given the context paragraph and prior questions.

We can use a standard fully connected layer in the footsteps of Ju et al. [13] but it makes more sense to leverage the capabilities of LSTMs as they can handle context better. Thus, we pass the concatenated output from the embeddings to a trainable LSTM or fully connected generator in this step. We then use a softmax layer to calculate the output probability distribution for the tokens generated from the model. From this, we take the token with the highest probability of being the start token. Similarly we train an LSTM or a fully connected layer to generate the token for the end of the answer span.

Based on this, we can generate the total answer span using Gaussian noise as inputs.

6.1.3 The Discriminator

The discriminator shares the contextual embeddings with generator. However, the discriminator receives the answer embeddings conditioned on the question and context embeddings, leading to a conditional GAN setting. It is conceptualised as an LSTM model that has a softmax layer that selects the output having the highest probability of belonging to the category provided, i.e. in our case whether a given set of answer tokens are conditionally acceptable given the question and the context.

The target of the discriminator is to differentiate between the start/end tokens from the generator and the start/end tokens from the real dataset. We also add some random noise to the discriminator inputs. This will make it more difficult for the discriminator to train, while requiring the generator to develop a more sophisticated latent space.

6.1.4 Conditional GAN setting

I have described our generator and discriminator. We now focus on training the conditional GAN model that has been proposed. I have detailed the basic architecture in 5.

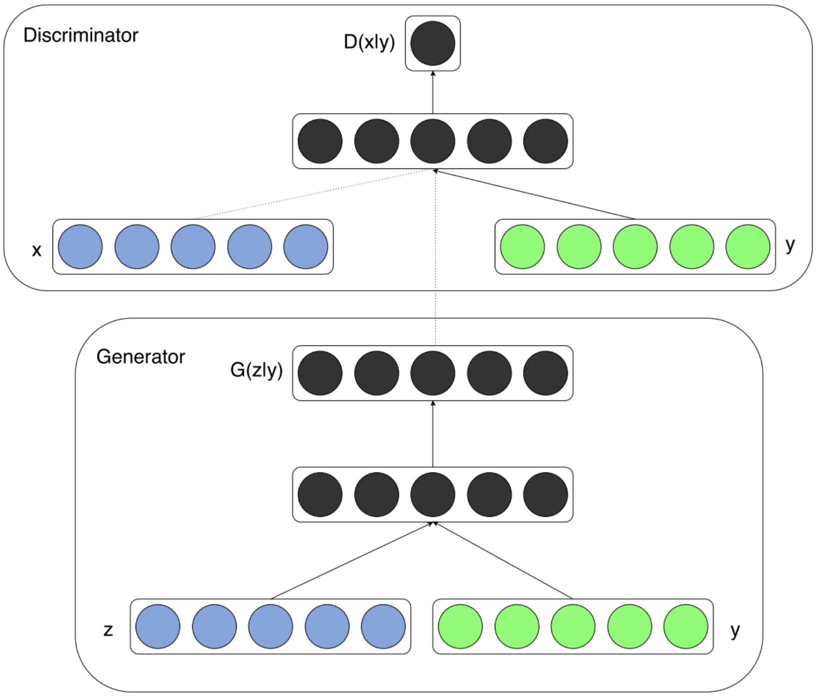

Figure 5: The Conditional GAN architecture. The generator models the data based on the input and the context encoding, and the discriminator, unlike the basic GAN architecture, discriminates based on the condition provided, instead of the Boolean real/fake values. [17]

Next we look at the loss function we will be minimising. We will not be considering the traditional conditional GAN loss as it is notoriously unstable, prone to mode collapse, and does not guarantee high quality outputs. We will, instead, be using the least-squares GAN loss as described below:

$$min_D V_{LSGAN}(D) = \frac{1}{2}\mathop{\mathbb{E}}_{x,y}[(D(x,y)-1)^2] + \frac{1}{2}\mathop{\mathbb{E}}_{z\sim p_z(z)}[(D(G(z)))^2] \tag{3}$$

$$min_G V_{LSGAN}(G) = \frac{1}{2}\mathop{\mathbb{E}}_{z\sim p_z(z)}[(D(G(z))-1)^2] \tag{4}$$

Unlike the traditional GAN loss as defined in Section 4.1, we can see that the cGAN discriminator loss is conditioned on the conditioning variable y, which means that the generated outputs will depend on it too. These losses will be set in a minimax game in the regular way. We now modify it to use it in our case. We replace the input and conditioning variables with the ones we will be using.

$$min_D V_{LSGAN}(D) = \quad \frac{1}{2} \mathop{\mathbb{E}}_{\mathbf{a},\ \mathbf{Q},\ \mathbf{A},\ \mathbf{C}} [(D(\mathbf{a},\ \mathbf{Q},\ \mathbf{A},\ \mathbf{C}) - 1)^2] + \frac{1}{2} \mathop{\mathbb{E}}_{z \sim p_z(z)} [(D(G(z)))^2] \tag{5}$$

$$min_G V_{LSGAN}(G) = \quad \frac{1}{2} \mathop{\mathbb{E}}_{z \sim p_z(z)} [(D(G(z)) - 1)^2] \tag{6}$$

where $\mathbf{a}$ is the embedding for the current answer, $\mathbf{Q}$ is the question history embedding, $\mathbf{A}$ is the answer history embedding, and $\mathbf{C}$ is the context embedding.

6.2 VAE-based model for CoQA

Next we look at the variational autoenconder based model for our task. The conditional VAE that we are considering has been inspired by Wang et al [27]. It consists of an encoder and a decoder coupled with a latent space consisting of a *prior net*, a *posterior net*, and a latent variable z that will be optimised on generating coherent, acceptable responses to the questions.

6.2.1 Encoder

We use a similar contextual word embedding encoder for the VAE as we were for the GAN-based model. We add a LSTM to generate the embeddings of the context. We want to ensure that the encoder encodes data in a way that the posterior can easily be modelled by the decoder. To achieve this end, we import KL divergence constraint upon the LSTM such that $KL(q(z|\mathbf{a},\ \mathbf{C},\ \mathbf{Q},\ \mathbf{A})||p(z|\mathbf{C},\ \mathbf{Q},\ \mathbf{A})$ is kept within a certain range. We expand upon this definition in section 6.2.3. We can also use some scheduled sampling techniques as proposed by Bengio et al. [5]

6.2.2 Decoder

Next we look at the decoder for our VAE model. We can choose a simple decoder that can generate the span of the answer text, given the input from the latent space, and the contextual word embeddings. We use an LSTM architecture similar to the encoder. We further connect the encoder and decoder via attention in the intermediate layers of both so as to provide the decoder with more sophisticated representations. Here, we stick to using the latent variable z to initialise the states of the decoder.

6.2.3 Conditional VAE Training

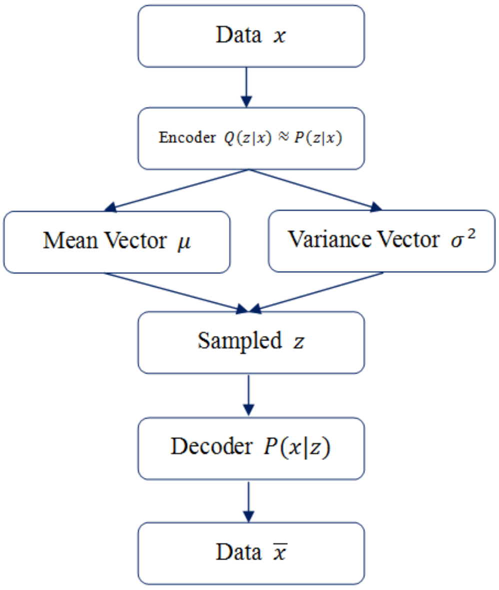

Figure 6: The conditional variational autoenconder architecture [3]. We use the question, the prior questions, and the prior answers as the conditions on which the current answer should be conditioned.

We have described the general conditional VAE architecture above. We train the conditional VAE by optimising the evidence lower bound (ELBO) which maximises the probability of the answer, given the entire context as shown below.

$$logp(\mathbf{a}|\mathbf{C},\mathbf{Q},\mathbf{A}) \geq \underset{q(z|\mathbf{a},\ \mathbf{C},\ \mathbf{Q},\ \mathbf{A})}{\mathbb{E}}(logp(\mathbf{a}|\mathbf{C},\ \mathbf{Q},\ \mathbf{A},z) - KL(q(z|\mathbf{a},\ \mathbf{C},\ \mathbf{Q},\ \mathbf{A})||p(z|\mathbf{C},\ \mathbf{Q},\ \mathbf{A}) \tag{7}$$

Here q is an approximated posterior, and p is the conditional output distribution. We will use KL annealing to increase the weightage of the KL term as training continues to proceed

7 Discussion

A point worth noting in the paper is that I have not performed experiments on the models proposed. While performing them would have provided a stronger paper, the lack of computational power required for the task was something I could not get past. I would, nonetheless, relish the opportunity to do so.

We defined both models as conditional models, requiring the question, history, and context paragraphs as the information upon which the answer would be conditioned. This makes intuitive sense as in the case of humans, we condition our responses on contextual cues, the current question, and the conversation we have had with a person so far. This is, however, not the only approach we can consider as we saw with the models that motivated our approach. We can use a dual-VAE, or an adversarial training process in a RL setting. Similarly we can use an adversarial VAE to perform the task, combining the capabilities of both the GAN and the VAE.

We also need to address the benefits of using generative modelling. One of the most desirable properties is that these models learn a distribution of the answers conditioned on the relevant variables, from which answers can be generated. This, because of its richer capacity to represent data as compared to discriminative models, can lead to more coherent and varied answers. Since the model is not trying to maximise the probability of an answer given the conditioning information, but in fact, has access to a data distribution, it is less likely to output ”I don’t know” as a high probability answer very often.

An interesting manner in which our models departed from the discriminative models for CoQA is that they do not have a layer that predicts ”yes”, ”no”, or ”unknown” answers. I am of the opinion that because the model learns the data distribution, it should be able to predict those answers from the information it has, instead of having an explicit layer for it. This is, again, an assumption, but one supported by the literature we have mentioned. If this is an assumption we can show as true, we can potentially use it in other NLP use-cases to reduce model complexity.

Lastly, we can compare the two proposed models. They have both been made, keeping in consideration SOTA generative models on various NLP tasks. We cannot, without experiments, compare the quality of their outputs. However, a metric we can compare them on is their complexity. The VAE model has a simpler structure with no discriminator network, and the GAN is relatively more heavily engineered. This should make the VAE train faster given the same dataset.

8 Conclusion

In conclusion, I have discussed the Conversational Question Answering dataset and provided a basic notion of the kind of models that have proven effective on it. I have then proposed two models in the generative modelling regime, one based on a conditional GAN, and the other based on a conditional VAE. Their architectures have been discussed briefly, and then we have expanded upon their loss functions to give a notion of how these models can be trained. Lastly, I have provided some analysis on the models, their benefits, and other models we can consider from the same space.

References

[1] A beginner’s guide to generative adversarial networks (gans). https://pathmind.com/wiki/generative-adversarial-network-gan. Accessed: 2020-04-20.

[2] From autoencoder to beta-vae. https://lilianweng.github.io/lil-log/2018/08/12/from-autoencoder-to-beta-vae.html#vae-variational-autoencoder. Accessed: 2020-08-06.

[3] Vae and conditional vae. https://khhuang.net/blog/machine%20leaning/2017/06/26/vae-and-conditional-vae.html. Accessed: 2020-08-20.

[4] Pierre Baldi. Autoencoders, unsupervised learning, and deep architectures. In Isabelle Guyon, Gideon Dror, Vincent Lemaire, Graham W. Taylor, and Daniel L.

Silver, editors, *ICML Unsupervised and Transfer Learning*, volume 27 of *JMLR Proceedings*, pages 37–50. JMLR.org, 2012.

[5] Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. Scheduled sampling for sequence prediction with recurrent neural networks, 2015.

[6] Leo Breiman and Nong Shang. Born again trees. *University of California, Berkeley, Berkeley, CA, Technical Report*, 1:2, 1996.

[7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[8] Jianfeng Gao, Michel Galley, and Lihong Li. *Neural Approaches to Conversational AI: Question Answering, Task-oriented Dialogues and Social Chatbots.* Now Foundations and Trends, 2019.

[9] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[10] B.F. Green, A.K. Wolf, C. Chomsky, and K. Laughery. Baseball: An automatic question answerer. In *Proceedings Western Computing Conference*, volume 19, pages 219–224, 1961.

[11] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks, 2016.

[12] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016.

[13] Ying Ju, Fubang Zhao, Shijie Chen, Bowen Zheng, Xuefeng Yang, and Yunfeng Liu. Technical report on conversational question answering. *arXiv preprint arXiv:1909.10772*, 2019.

[14] Phillip Isola Alexei A. Efros Jun-Yan Zhu, Taesung Park. Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks. *CVPR*, 2018.

[15] Oleksandr Kolomiyets and Marie-Francine Moens. A survey on question answering technology from an information retrieval perspective. 181(24):5412–5434, 2011.

[16] Jiwei Li, Will Monroe, Tianlin Shi, Sébastien Jean, Alan Ritter, and Dan Jurafsky. Adversarial learning for neural dialogue generation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2157–2169, 2017.

[17] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets, 2014.

[18] Yasuhito Ohsugi, Itsumi Saito, Kyosuke Nishida, Hisako Asano, and Junji Tomita. A simple but effective method to incorporate multi-turn context with bert for conversational machine comprehension. *arXiv preprint arXiv:1905.12848*, 2019.

[19] Pranav Rajpurkar, Robin Jia, and Percy Liang. Know what you don't know: Unanswerable questions for squad. *arXiv preprint arXiv:1806.03822*, 2018.

[20] Siva Reddy, Danqi Chen, and Christopher D Manning. Coqa: A conversational question answering challenge. *Transactions of the Association for Computational Linguistics*, 7:249–266, 2019.

[21] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to sequence learning with neural networks, 2014.

[22] Van-Khanh Tran and Minh Le Nguyen. Dual latent variable model for low-resource natural language generation in dialogue systems. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pages 21–30, 2018.

[23] A. Viswanathan, B. Mehta, M. P. Bhavatarini, and H. R. Mamatha. Text to image translation using generative adversarial networks. In *2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pages 1648–1654, 2018.

[24] Ellen M. Voorhees and Dawn M. Tice. Building a question answering test collection. In *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '00, page 200–207, New York, NY, USA, 2000. Association for Computing Machinery.

[25] Shengxian Wan, Yanyan Lan, Jiafeng Guo, Jun Xu, Liang Pang, and Xueqi Cheng. A deep architecture for semantic matching with multiple positional sentence representations, 2015.

[26] Bingning Wang, Kang Liu, and Jun Zhao. Conditional generative adversarial networks for commonsense machine comprehension.

[27] Tianming Wang and Xiaojun Wan. T-cvae: Transformer-based conditioned variational autoencoder for story completion. In *IJCAI*, pages 5233–5239, 2019.

[28] Robert Wilensky, David N. Chin, Marc Luria, James Martin, James Mayfield, and Dekai Wu. The Berkeley Unix Consultant project. *Computational Linguistics*, 14(4), 1988.

[29] Liu Yang, Qingyao Ai, Jiafeng Guo, and W. Bruce Croft. anmm: Ranking short answer texts with attention-based neural matching model, 2018.

[30] Lei Yu, Karl Moritz Hermann, Phil Blunsom, and Stephen Pulman. Deep learning for answer sentence selection, 2014.