Unsupervised Hierarchical Model for Deep Empathetic Conversational Agents

Vincenzo Scotti

https://orcid.org/0000-0002-8765-604X

DEIB, Politecnico di Milano
Via Golgi 42, 20133, Milano (MI), Italy
vincenzo.scotti@polimi.it

Abstract

Empathy is a fundamental mechanism that characterises human beings and affects how they interact each other. Implementing empathetic intelligence within conversational agents is thus necessary to make them appear more "human", with the final goal of improving the user experience. Empathy is related to the emotional sphere, but also affects how we understand and share the feelings of another. To embed these concepts into a conversational agent, we suggest to leverage the hierarchical structure of human language; casting empathy as a control problem over high-level conversation's aspects allows the agent to reason and plan to act empathetically within the conversation. In this chapter we present the architecture of an open-domain empathetic conversational agent. We trained the agent in two steps. In the first step, the agent learns the relevant high-level structures of the conversation, leveraging a mix of unsupervised and supervised learning. Then, in the second step, we refine the agent through supervised and reinforcement learning, making it able to elicit positive sentiments in the user. We based the whole architecture on a *Seq2Seq* Deep Neural Network, which directly generates the response tokens. Results from automatic metrics show promising scores that open the possibility for a human-based evaluation on real-time conversations.

Keywords: Natural Language Processing, Deep-Learning, Conversational agent, Reinforcement learning, Transformers network, Empathetic computing.

1. Introduction

The recent advances in Artificial Intelligence (AI) powered by Deep Learning (DL) significantly pushed forward the state-of-the-art in many fields [1]. In particular, the Seq2Seq transformer networks highly influenced Natural Language Processing (NLP) [2] Transformers are deep neural networks designed to deal with sequential data, as text streams, and capture and exploit (long term) sequential relations.

Such transformer networks can be easily fine-tuned into open-domain chatbots [3], and can be used for both retrieval and generative models. The formers select the response from a pool of available ones, while the latters generate the sequence of tokens composing the response [1]. Generative models can adapt better to unseen situations since their responses are not limited to the reference pool adopted by retrieval-based models.

This paper is interested in a subset of the open-domain chatbots, called empathetic chatbots [4] [5]. These chatbots are designed and built according to models and principles of empathy, a fundamental mechanism for human interactions [6] [7]. Proper implementation of such mechanism would be an essential step towards more human-like chatbots, tightening the gap between humans and machines.

Our approach is based on Seq2Seq networks and aims at proposing a viable solution to empathetic chatbots. Moreover, we threat empathy as a control problem using a reinforcement learning approach. We train the chatbot to leverage a self-learnt high-level dialogue structure for planning conversational acts that maximise the reward needed by the reinforcement learning approach.

We divide this chapter into the following sections. In Section 2, we briefly present the latest results on neural chatbots, and the current approaches for empathetic chatbots. In Section 3, we explain how we deal with empathy in out chatbot. In Section 4, we present the architecture and training procedure of the chatbot's underlying neural network. In Section 5, we explain how we evaluated our chatbot and present the evaluation results. In Section 6, we summarise our work and provide hints about possible future works.

2. Related works

In this section, we present a brief recap of the latest approaches for chatbots based on Seq2Seq neural networks, and an overview of current solution for emapthetic chatbots.

2.1. Seq2Seq chatbots

Deep Neural Networks for sequence analysis (i.e., the Seq2Seq models) enabled the design of retrieval and generative chatbots with incredible capabilities [8] [9] . Retrieval chatbots rely on a corpus of possible dialogue turns to predict their responses. They do not suffer from disfluencies[1], but they lack flexibility as they are limited to the set of available responses into the pool [10]. On the other hand, generative models are more flexible, as they can also produce plausible responses in front of unseen contexts [4] [11] [12], but are prone to disfluencies.

There exist hybrid solutions combining the two approaches; Retrieve-and-refine models generate starting from the retrieved response, used as an example [13], while Multitask models, instead, have both retrieval and generative capabilities in a single architecture, trained concurrently [3].

In the last ten years, the approach evolved from plain Seq2Seq causal models [14] [15] towards more complex hierarchical models, leveraging either continuous [16] [17] [18]or discrete hidden representations [19].

[1] In this context, the term “disfluence” means the generation of meaningless sentences.

These early deep learning solutions were realised through recurrent neural networks. However, such networks were limited by the sequential analysis approach (which made it impossible to parallelise the computation) and the inability to manage extended contexts (due to the degradation of the hidden representation). Thus, current approaches rely on attention mechanisms and transformer architectures [2].
Thanks to the high availability of pre-trained transformers [20] it is now possible to fine-tune them into conversational agents without the need for long training sessions on huge corpora, while still achieving impressive results [3]. These chatbot models are able to capture complex long-term relationships (and hence longer contexts) and allow for completely parallel computation [11] [12] [21] [22].
Usually, these neural chatbots are trained with a supervised learning approach. Given the context (i.e., the considered conversation history), retrieval models are trained to maximise the posterior probability of the whole target response. Instead, generative models are trained to maximise the log-likelihood of the next token in the response, given the context and the preceding response tokens (the autoregressive approach).
The training approach is not limited to the supervised one: it is possible to rely on a reinforcement learning approach, and indeed many solutions have been proposed to train open-domain agents through reinforcement learning [23] [24] [25]. Unlike task-oriented chatbots, however, in the case of open-domain chatbots the reward is not well defined. Thus, metrics measuring social conversational skills and conversation goodness are used as rewards in the reinforcement learning problem. Various solutions were proposed to measure such aspects, even through learnt metrics [26].

2.2. Empathetic chatbots

As premised, empathetic conversational agents are a subclass of open-domain chatbots; in particular, such agents try to perceive emotions and react to them showing empathy, a fundamental mechanism of human-human interaction. Empathy can be roughly described as the ability to understand another's inner state and, possibly, respond accordingly (more on this in Section 3).
In the last years, a growing interest in this area led to several solutions being proposed to implement empathy in conversational agents. XiaoIce represents an impressive example of an empathetic agent [4]. It implements both emotional and cognitive aspects of empathy, and is powered by knowledge grounding, persona grounding, and image grounding. Moreover, it is deployed on many social media, thus having access to users' profiles for a more personalised experience (in fact, it is possible to mine useful personal information from websites like Twitter of Facebook [27]). Additionally, it is embodied through voice and an avatar, making it easier to perceive the agent as a human.
The agent embodiment through visual and voiced interaction modules, although quite powerful in improving the user experience, is hard to manage. Thus, other solutions limit the interaction to text exchanges, yielding a more straightforward development process. Conversational agents like CAiRE [5], MoEL [28] and EmpTransfo [29] implement empathy in their ability to recognise the user's emotion or predict the most appropriate response emotion, dialogue act and more. These agents learn the emotional mechanism, generating conditioned text; in other words, they have the ability to generate a response given some high-level attributes, like the desired emotion.
Such textual models learn to simulate empathy by imitating good empathetic behaviour examples, but it is possible to go beyond this approach: setting an explicit objective compatible with an empathetic behaviour makes it possible to implicitly train the agent towards an empathetic behaviour.
The idea is to set an objective that implicitly requires the agent to understand the user's inner state from the conversation context, and to act accordingly. In particular, it is the case of agents

having as as target to elicit a positive sentiment in the user. Agents like Emo-HRED [30] or MC-HRED [31] select the desired high-level response attributes (emotion and dialogue act) to maximise such target. Thus, the actual response generation is conditioned on the selected attributes. This approach is also directly applicable at a lower dialogue level, like in the sentiment look-ahead network [32]. This network leverages reinforcement learning to alter the probability distribution of the next token in the response, maximising the positivity of the user's expected sentiment.

3. Approach to empathy

Empathetic computing is a generalisation of affective computing [33]. Early works on affective computing explained how a machine would not be completely intelligent as long as it does not perceive the user's emotions. Empathy completes this concept by explaining how it is important to show emotional and cognitive intelligence [6]. These aspects of empathy allow understanding someone's mental state (like emotion or intent).

Empathy affects human interactions at different levels, as in a hierarchy, and multiple frameworks reflect that [6] [7]. In this work, we propose following this same approach, and building the chatbot to approach the conversation from a hierarchical perspective. The idea is to treat empathy as a control problem and have the agent selecting a high-level abstract repose first, and then yielding the low-level response, all according to an empathetic policy. This policy controls the empathetic behaviour of the agent [34].

We relied on data-driven approaches to build our generative empathetic chatbot, following the impressive advances observed in open-domain agents [4] [11] [13] [12] [21]. Our approach started from a probabilistic language model; in particular, we used a pre-trained language model to have strong initialisation in features and generative capabilities, and the we fine-tuned it into the final dialogue language model.

Unlike previous works on empathetic agents, we propose leveraging unsupervised learning to extract a discrete high-level dialogue model during the dialogue language modelling training. Previous works rely on high-level labels (like emotion or dialogue act) available on annotated corpus [28] [5] [29]; this may represent a limitation since not all corpora are annotated, or are based on the same label set. Our approach allows merging multiple corpora and thus training a more complex model with possibly better generative capabilities.

We further fine-tuned the agent using reinforcement learning on an empathetic objective to provide the agent with empathetic capabilities. We refined the agent to maximise the user's positive sentiment (extracted from the next conversation turn) and the user's engagement (measured as the next turn relative length, with respect to the previous one). This step is necessary for the agent to learn the aforementioned empathetic policy. We use a discrete high-level model and an hybrid training framework to ensure this refinement step doesn't break the agent's conversational capabilities [19].

4. Chatbot implementation

In this section, we describe the probabilistic language model we used to implement our dialogue agent, and the training process we followed to embed the agent with empathy[2].

[2] The code base with the dialogue agent model and the training process are available at https://github.com/vincenzo-scotti/dldlm/tree/v2.0

4.1. DLDLM architecture

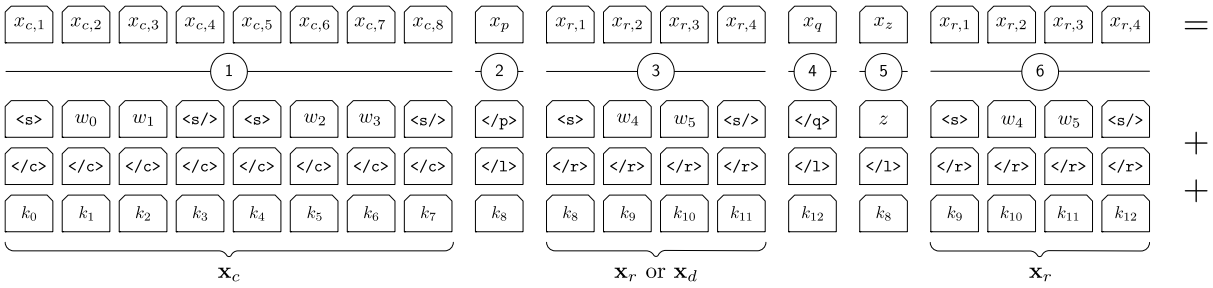

Figure 4.1. Input structure: top row elements are the actual input embeddings to the hidden transformation, the second row elements are token embeddings, the third row elements are token type embeddings, while the fourth row elements are position embeddings. The w_i tokens were identified by the original GPT-2 tokeniser. Numbers in circles identify the steps in input processing.

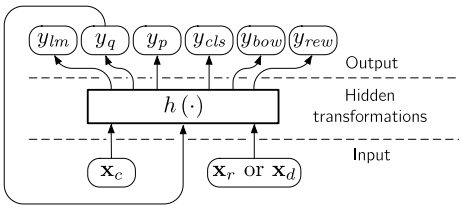

Figure 4.2.a. Training.

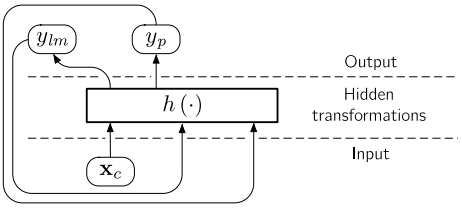

Figure 4.2.b. Inference.

Figure 4.2. Model abstract architecture and Input/Output flow.

As premised, we build our agent through a probabilistic dialogue language model; in particular, we designed and implemented it starting from the well-known GPT-2 [35] language model. Then, we extend the resulting model to include the hierarchical aspects of language we want to learn and exploit.

We extended the vanilla Seq2Seq architecture of GPT-2 with additional heads (i.e., final linear transformations). The idea was to learn a set of discrete latent codes by clustering the responses while learning to predict them. In doing that, we follow an approach similar to PLATO [21] [22]; however, our approach also predicts latent codes, while PLATO only uses posterior recognition.

The resulting dialogue language model is a variational auto-encoder with discrete latent codes. The model learns the latent codes in an unsupervised way and uses the recognised or predicted latent codes (at train and inference time, respectively) to condition the response generation. We call this architecture Discrete Latent Dialogue Language Model (DLDLM).

The model takes the sequence of context tokens $\boldsymbol{x_c}$ and the sequence of response tokens as input. The response tokens can be either those of the correct response $\boldsymbol{x_r}$ or a distractor $\boldsymbol{x_d}$ (for

multiobjective training; more on this in Section 4.2). During training, the model's input comprises the entire sequence of response tokens. At inference time, instead, the response tokens are generated in an autoregressive fashion. The overall input structure is presented in Figure 4.1.

The model fetches three kinds of embeddings that sum together at each position in the sequences, to encode the input and feed the hidden transformations. We distinguish among token embeddings, token type embeddings and position embeddings.

Token embeddings are the regular embeddings calculated from the textual sequence. We wrap each turn with special token embeddings to indicate the beginning (**<s>**) and end (**<s/>**) of each of them. We also introduce additional embeddings to encode the latent codes z. Finally, we have special tokens to instruct the model to perform the posterior (**</q>**) or prior (**</p>**) latent analysis.

Token type embeddings are three, and represent where the tokens come from: context (**</c>**), response (**</r>**) or latent analysis (**</l>**). Finally, we use position embeddings to encode positional information into the token representation.

On top of the hidden transformations $h(\cdot)$, the model has six distinct heads:

- We use the Language Modelling head $y_{lm}(\boldsymbol{x}_c, z, \boldsymbol{x}_r)$ to predict the probability of the next response token: $P(\boldsymbol{x}_{r,i}|\boldsymbol{x}_c, z, \boldsymbol{x}_{r,j<i})$;
- We use the Latent Posterior head $y_q(\boldsymbol{x}_c, \boldsymbol{x}_r)$ to predict the posterior latent distribution $P(z|\boldsymbol{x}_c, \boldsymbol{x}_r) \stackrel{\text{def}}{=} \mathrm{Q}$;
- We use the Latent Prior head (or Policy head) $y_p(\boldsymbol{x}_c)$ to predict the prior latent distribution $P(z|\boldsymbol{x}_c) \stackrel{\text{def}}{=} \mathrm{P}$;
- We use the Classification head $y_{cls}(\boldsymbol{x}_c, \boldsymbol{x}_r)$ to predict the posterior probability that a given response is correct $P(\mathcal{C} = \text{correct}|\boldsymbol{x}_c, \boldsymbol{x}_r)$, and also the posterior probability that a given distractor response is wrong $P(\mathcal{C} = \text{wrong}|\boldsymbol{x}_c, \boldsymbol{x}_d) = 1 - P(\mathcal{C} = \text{correct}|\boldsymbol{x}_c, \boldsymbol{x}_d)$;
- We use the Bag-of-Words (BoW) head $y_{bow}(\boldsymbol{x}_c, z)$ to predict the normalised BoW representation of the response $\mathrm{BoW}(\boldsymbol{x}_r) = y_{bow}(\boldsymbol{x}_c, z)$;
- We use the Reward head $\hat{\boldsymbol{r}} = y_{rew}(\boldsymbol{x}_c, z)$ to predict the immediate reward $\boldsymbol{r}$.

The heads are used differently, depending on whether the model is deployed at train or inference time, as depicted in Figure 4.2 (more on this in Section 4.2).

Following the number notation Figure 4.1, the model follows this pipeline:

1. The model encodes $\boldsymbol{x}_c$ into the encoded context $\boldsymbol{H}_c = h(\boldsymbol{x}_c)$ using the hidden transformations.
2. The model encodes the prior latent analysis token **</p>** into $\boldsymbol{h}_p = h(\boldsymbol{x}_p, \boldsymbol{H}_c)$, given the encoded context $\boldsymbol{H}_c$, and predicts the prior probability distribution P using $y_p(\cdot)$.
3. The model encodes $\boldsymbol{x}_r$ into the encoded response $\boldsymbol{H}_r = h(\boldsymbol{x}_c, \boldsymbol{H}_c)$, given the encoded context $\boldsymbol{H}_c$. During training, $\boldsymbol{x}_d$ is encoded too, as an alternative path to $\boldsymbol{x}_r$.
4. The model encodes the posterior latent analysis token **</q>** into $\boldsymbol{h}_q = h(\boldsymbol{x}_q, \boldsymbol{H}_c, \boldsymbol{H}_r)$, given the encoded context $\boldsymbol{H}_c$ and response $\boldsymbol{H}_r$, and predicts the posterior probability distribution Q using $y_q(\cdot)$. Then, the model computes the posterior probability of a response to be the correct one (for both $\boldsymbol{x}_r$ and $\boldsymbol{x}_d$), on top of $_q$, using the retrieval head $y_{cls}(\cdot)$.
5. The model encodes the selected high-level latent token z into $\boldsymbol{h}_z = h(z, \boldsymbol{H}_c)$ from the encoded context $\boldsymbol{H}_c$. During training, z is sampled from Q; during inference, from P.

Then, on top of $\boldsymbol{h}_z$, the model predicts the expected reward with $y_{rew}(\cdot)$, and predicts the BoW representation with$y_{bow}(\cdot)$.

6. Finally, the model computes the posterior probability of the next token $x_{r,i}$, given the encoded context $\boldsymbol{H}_c$, the encoded latent $\boldsymbol{h}_z$, and the preceding response tokens $x_{r,j<i}$, using the language modelling head $y_{lm}(\cdot)$.

4.2. Training

We train the DLDLM model in two steps.

During the first step, the model learns the high- and low-level dialogue model; we leveraged unsupervised learning to extract the high-level model and supervised learning to extract the low-level dialogue model.

During the second step, the model learns the empathetic behaviour; we leveraged a hybrid reinforcement and supervised learning approach to learn the empathetic policy without breaking the underlying dialogue model.

4.2.1. Discrete latent dialogue model

We trained the network on the first step using mini-batches X of dialogue response samples. Each sample is a quadruple composed of a sequence of context tokens $\boldsymbol{x}_c$, a sequence of response tokens $\boldsymbol{x}_r$, a sequence of distractor tokens $\boldsymbol{x}_d$ and an immediate reward vector r; we update the parameters Θ of the network to minimise the loss described in the following equation.

$$\mathcal{L}(X;\Theta) = \mathbb{E}_X[\mathcal{L}_{LM}(\boldsymbol{x}_c,\boldsymbol{x}_r)] + \mathbb{E}_X[\mathcal{L}_{KLt}(\boldsymbol{x}_c,\boldsymbol{x}_r)] + \mathbb{E}_X[\mathcal{L}_{CLS}(\boldsymbol{x}_c,\boldsymbol{x}_r,\boldsymbol{x}_d)] + \mathbb{E}_X[\mathcal{L}_{BoW}(\boldsymbol{x}_c,\boldsymbol{x}_r)] + \sqrt{\mathbb{E}_X[\mathcal{L}_{REW}(\boldsymbol{x}_c,\boldsymbol{x}_r,\boldsymbol{r})]}$$

where:

- $\mathcal{L}_{LM}(\cdot)$ is the average negative log-likelihood of observing the response tokens given the context tokens and the preceding response tokens (i.e., the usual language modelling loss) to train the dialogue language model;
- $\mathcal{L}_{KLt}(\cdot)$ is the thresholded Kullback-Leibler (KL) divergence of P(·) from Q(·), used to prevent the vanishing KL issue [36] and train the discrete latent model [37].
- $\mathcal{L}_{CLS}(\cdot)$ is the contrastive binary cross-entropy to train the retrieval model;
- $\mathcal{L}_{BoW}(\cdot)$ is the average negative log-likelihood of the response tokens computed from z, to help training the latent model;
- $\mathcal{L}_{REW}(\cdot)$ is the mean squared reward prediction error, to help modelling the hidden features for the following training step.

During this training step, we modify the activation of the posterior head $y_q(\cdot)$. We employ a gumbel-softmax(·) [38] instead of the regular softmax(·) as, during training, we are interested in dealing with a distribution as close as possible to the categorical one (due to the discrete approach), while still needing to maintain the latent sampling process differentiable.

4.2.2. Empathetic policy

During the second step, we trained the network using mini-batches X of episodes E (i.e., entire dialogues). Then, for training the empathetic controller (i.e., the empathetic policy) we resorted to a policy gradient algorithm: REINFORCE [39]. In particular, we used the off-policy version of the algorithm, to avoid wasting resources for conversation simulations, and to avoid introducing errors due to the possible faults in the dialogue generation process (sometimes, models like the one we are designing tend to yield dull or inconsistent responses [8]).

To avoid breaking the generative capabilities learnt from the previous step, we resorted to a hybrid reinforcement and supervised training objective [32], to maximise the hybrid objective function described in the following equation. The two objectives are weighted by a parameter

$\lambda \in [0,1] \subseteq \mathbb{R}$ to control the trade-off between the reinforcement learning objective $J_{RL}(E)$ and the supervised learning loss $\mathcal{L}_{SL}(\cdot)$, in the hybrid training.

$$\mathcal{J}(E;\Theta) = \lambda\mathbb{E}[J_{RL}(E)] + (1-\lambda)\mathbb{E}[\mathcal{L}_{SL}(E)]$$

$$J_{RL}(E;\Theta) = -\sum_{t=1}^{|E|} \tilde{G}^{(t)} \cdot (\mathcal{L}_{NLLz}\left(\boldsymbol{x}_c^{(t)}, \boldsymbol{x}_r^{(t)}\right) + \alpha\mathcal{L}_{LM}\left(\boldsymbol{x}_c^{(t)}, \boldsymbol{x}_r^{(t)}\right))$$

where:

- $J_{RL}(\cdot)$ is the reinforcement learning objective to maximise, computed as in the previous equation;
- $\mathcal{L}_{SL}(\cdot)$ is the supervised learning loss to minimise, defined as in first equation, but with $\mathcal{L}_{NLLz}(\cdot)$ instead of $\mathcal{L}_{KLt}(\cdot)$ (see the next point);
- $\mathcal{L}_{NLLz}(\cdot)$ is the negative log-likelihood of predicting the latent code that maximises Q(·) using P(·).
- $\alpha \in \{0,1\}$ is a parameter to control whether to use the REINFORCE objective to influence also the low level language modelling ($\alpha = 1$) or only the high-level policy ($\alpha = 0$).
- $\tilde{G}^{(t)}$ is a standardised cumulative discounted reward computed under the behaviours policy at time step t.

As from previous step we resorted to the gumbel-softmax(·).

4.2.3. Hyperparameters

We trained and refined two versions on the network based on the 117 and 345 million parameters versions of the original GPT-2.

The two models were trained for 30 and 10 epochs, respectively, during the first training step, and for a single epoch in the second one. During the first training step we used a mini-batch size of 64, while during the second training step a mini-batch of size 1 (a single episode) was used.

In each context-response pair we considered only contexts up to 256 tokens and responses up to 128 tokens. We leveraged the original GPT-2 tokeniser to encode the turn strings.

About the training process, we used the AdamW optimiser [40] and, in all training processes, we adopted a linear learning rate schedule with 0.2% of update steps warmup. The maximum learning rates in the two implementations were $6.25 \cdot 10^{-5}$ and $3.125 \cdot 10^{-5}$, respectively.

Finally, the gumbel-softmax(·) used a temperature rescoring of T = 2/3.

5. Evaluation

In this section, we present the approach we followed on the evaluation of the agent, the corpora we employed, and the subsequent results.

5.1. Corpora

We trained and evaluated our chatbot on a mix of different well-curated corpora to have sufficient data to extract a reliable high-level model. In particular, we merged four different open-domain conversation corpora: DailyDialog (DD) [41], EmpatheticDialogues (ED) [42], Persona-Chat (PC) [43], and Wizard of Wikipedia (WoW) [44]. We used the same splits of the original corpora to collect the train and validation samples we used in the learning steps, and the test samples we used in the evaluation steps. Table 1 reports the main statistics about the corpora.

Table 1 - Main statistics on the considered corpora organised per split

	Train			**Validation**			**Test**		
	Dialogues	**Turns per dialogue**	**Tokens per turn**	**Dialogues**	**Turns per dialogue**	**Tokens per turn**	**Dialogues**	**Turns per dialogue**	**Tokens per turn**
DD	11118	7.84 ± 4.01	14.37 ± 10.83	1000	8.07 ± 3.88	14.28 ± 10.52	1000	7.74 ± 3.84	14.56 ± 10.92
ED	19533	4.31 ± 0.71	15.90 ± 9.80	2770	4.36 ± 0.73	17.08 ± 9.66	2547	4.31 ± 0.73	18.16 ± 10.38
PC	8939	14.70 ± 1.74	12.11 \pm 4.24	1000	15.60 ± 1.04	12.37 ± 4.05	968	15.52 ± 1.10	12.23 ± 4.00
WoW	18430	9.05 ± 1.04	19.88 ± 9.64	981	9.08 ± 1.02	19.89 ± 9.62	965	9.03 ± 1.02	19.91 ± 9.58
Total	58020	8.09 ± 3.99	15.97 ± 9.33	5751	7.77 ± 4.45	15.49 ± 8.81	5480	7.75 ± 4.45	15.76 ± 9.15

As premised, we considered two distinct rewards to maximise, in the empathetic learning step. The elicited sentiment reward was computed scaling the results of sentiment analysis of each turn, in a $[-1, 1] \subseteq \mathbb{R}$ range. The reward about the relative response length was computed as the difference between the number of next turn tokens and current response ones, normalising on the current response length; this reward was further scaled through a $\tanh(\cdot)$\$ to constrain the values in a $[-1, 1] \subseteq \mathbb{R}$ range. We leveraged an external tool (the SpaCy[3] library) to compute these values.

5.2. Approach

We evaluated the chatbot implementations through automatic metrics to assess the quality of the dialogue language model, and to assess the positive effects of the empathetic refinement. The 117M and 345M models were compared right after the first training step, after the policy fine-tuning, and after policy and language modelling joined the fine-tuning. In this way, we observed the effects of the various training steps.

To evaluate the generative capabilities of the dialogue language mode, we resorted to Perplexity (PPL) [3] [11] [12]. It is the most commonly used metric for this kind of evaluation. Moreover, it strongly correlates with human judgment on dialogue quality [12].

We maintain an off-policy approach to evaluate empathy and sociality, as in the training step [24]. Thus, we compute the average cumulative reward of the models, weighted on the probability of doing the same action under the baseline policy or the empathetic policies.

We split the evaluation in two parts to better observe the effects of fine-tuning for empathy, at different granularity levels.

[3] https://spacy.io

5.3. Results

Table 2 - Results of the PPL off-policy evaluation of the language model. Emp. (π) models refer to the policy fine-tuning, Emp. (π, LM) models refer to the policy and language modelling joined fine-tuning, the remaining models are the baselines (i.e., no fine-tuning). $r_{sent.} > 0$ *refers to the samples in the test set where the elicited sentiment reward is non-negative,* $r_{soc.} > 0$ *refers to the samples in the test set where the* $\tanh(\cdot)$ *of the elicited response relative length is non-negative.*

Model		**PPL**			
Configuration	**Size**	**All data**	$r_{sent.} > 0$	$r_{soc.} > 0$	$r_{sent.} > 0 \wedge r_{soc.} > 0$
Baseline	**117M**	18.27 ± 31.08	18.23 ± 31.76	17.22 ± 35.48	17.36 ± 36.73
	345M	14.67 ± 21.88	14.65 ± 22.32	13.93 ± 24.87	14.03 ± 25.85
Emp. (π)	**117M**	18.54 ± 33.51	18.46 ± 33.99	17.41 ± 38.28	17.51 ± 39.49
	345M	14.85 ± 21.80	14.80 ± 21.88	14.04 ± 24.18	14.11 ± 24.75
Emp. (π, LM)	**117M**	27.65 ± 70.93	27.42 ± 72.87	24.38 ± 79.75	24.60 ± 83.33
	345M	18.85 ± 62.54	18.90 ± 67.16	16.93 ± 32.81	17.03 ± 33.47

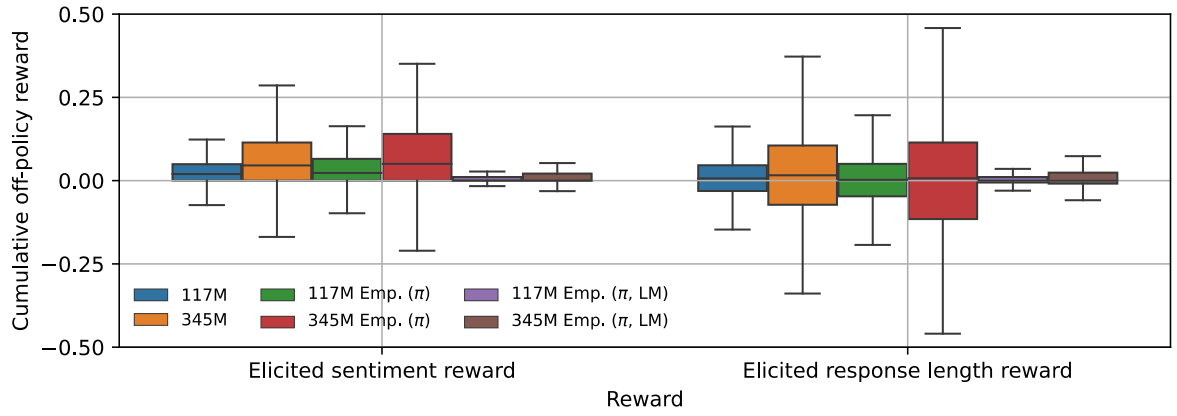

Figure 5.1. Results of the off-policy evaluation of the empathetic controller. Emp. (π) models refer to the policy fine-tuning, Emp. (π, LM) models refer to the policy and language modelling joined fine-tuning, the remaining models are the baselines (i.e., no fine-tuning).

We reported the results of the PPL evaluation\footnote{For the models that undergo empathetic fine-tuning on both policy and language modelling we got many infinite PPLs; to be able to compute these values we filtered all PPL > 10000 considering them as outliers (more comments about this later)} in Table 2 and the results of the empathetic policy (controller) in Figure 5.1. As premised, the reported results are from automatic metrics. A human evaluation should be

carried out to understand the actual chatbot behaviour in a better way. As for now, we limited the evaluation to this automatic approach to gathering early results on the proposed approach.
Concerning PPL, the first result we point out is that higher model complexity reflects in the results. The trained models achieved lower PPL scores when used in the 345M version. This lower PPL score reflects other results in literature, where authors showed how increasing model complexity does improve language modelling capabilities [11] [12].
Another point to highlight is how empathetic fine-tuning negatively affect language modelling capabilities. The "policy only" fine-tuning does not sensitively affect the PPL; this is expected since we did not alter the language modelling loss to train the model. The "joined" fine-tuning, however, produces way worse results. The 345M model ends up with a PPL closer to the 117M model of the other two tested configurations. Despite this being expected when computing PPL on the whole corpus, since the model needs to reject responses that may have negative rewards, we expected better results when considering only the subset of interactions with positive rewards (the last three columns of Table 2). Given the results of similar works [32], we expect that a better hyper-parameters search could help to fix this issue.
Finally, we would like to point out that the model always performs better on samples associated with positive rewards: it is correctly oriented towards responses that can promote users' positive sentiment and longer responses. Although a deeper analysis is required, the results we obtained can be a hint that our approach is viable to introduce an empathetic behaviour in conversational agents.
We immediately noticed two aspects of the weighted cumulative rewards value ranges concerning the empathetic policy. Empathetic fine-tuning of policy and language modelling leads to narrow ranges than other results. Moreover, the 345M models cover a more comprehensive range of values than the 117M ones.
Models that undergo fine-tuning of policy and language modelling achieve acceptable results in the evaluation of the weighted cumulative reward (as shown in Figure 5.1). The overall distribution of values is mostly non-negative, meaning that the model is oriented towards actions with non-negative rewards. This was the expected result of the fine-tuning. However, the narrow ranges indicate that instead of going toward positive rewards, the model learnt a "safe" policy where the rewards are close to 0. This behaviour is a common issue of off-policy learning. This result and the low PPL scores lead us to the realisation that this fine-tuning at multiple levels of granularity requires an ad-hoc analysis to work correctly, which we leave it as possible future work.
Models that undergo empathetic fine-tuning only on the policy, partially confirm the results from the PPL analysis. Observing the distributions of the cumulative elicited sentiment rewards, we notice that the model achieves higher maximum rewards and averages than the baseline counterparts. These higher scores mean that the empathetic fine-tuning positively affected the model towards a more empathetic behaviour, favouring the user's positive sentiment. Observing the cumulative distribution of elicited response's relative lengths, however, we do not find the same behaviour: maxima are higher than the baseline counterparts, but not averages. However, most of the distribution is non-negative, showing that the fine-tuning did not lead to undesired behaviours.
Finally, we point out that models that did not undergo empathetic fine-tuning still achieved good results in this evaluation. These results are primarily due to the corpus. Despite presenting examples of responses to cover both positive and negative rewards, there is an unbalance toward positive scores; thus, the model learns this behaviour directly from the training samples. Ideally, we would need a balanced corpus to have a sharper effect after fine-tuning; in practice, these data are hard to find, especially among well-curated dialogue corpora.
From these results, we evinced that the empathetic fine-tuning, limited to the high-level aspects of the conversation, achieves better results on elicited sentiment, showing a viable solution for

the development of empathetic chatbots. Moreover, acting only at a high level helps not to disrupt the language modelling capabilities of the agent (the difference in PPL between these models and the baseline counterparts can be considered negligible).

6. Conclusion and future work

This chapter described our solution to implement and train an empathetic chatbot using the a Seq2Seq approach. The agent is trained in a two-step process, starting from a pretrained probabilistic language model. During the first step, we fine-tune the agent to generate dialogue and learn a discrete latent dialogue structure. In the second step, we resort to hybrid reinforcement and supervised learning to exploit the dialogue structure and the dialogue generative capabilities, further refining the agent to optimise empathy-related rewards.
In our empathetic agent, we approach empathy as a control problem. We train and evaluate different versions of the Seq2Seq neural network in the experiments. The rewards we train the agent to optimise are the elicited positive sentiment (to enforce emotional intelligence) and the relative response length (to enforce a social behaviour that pushes the user towards openness). Applying the control at different levels of granularity, we observe that DLDLM produces better results when fine-tuned for empathy at the high-level dialogue model only.
As for now, we foresee two possible future directions. On one side, we are willing to refine the agent on more task-oriented conversations; the idea is to keep the open-domain conversation setting but with an overall goal requiring empathy and others' understanding, like in therapy or counselling sessions. On the other side, we are interested in completing the chatbot adding modules for voiced input/output, namely an Automatic Speech Recognition and a Text-to-Speech system. These extensions would make the agent appear more human and thus more relatable, a fundamental property for empathetic agents.

References

[1] D. Jurafsky e J. H. Martin, «Speech and language processing: an introduction to natural language processing, computational linguistics, and speech recognition (3rd Edition),» 2022.

[2] Q. Liu, M. J. Kusner e P. Blunsom, «A Survey on Contextual Embeddings,» *CoRR,* vol. abs/2003.07278, 2020.

[3] T. Wolf, V. Sanh, J. Chaumond e C. Delangue, «TransferTransfo: A Transfer Learning Approach for Neural Network Based Conversational Agents,» *CoRR,* vol. abs/1901.08149, 2019.

[4] L. Zhou, J. Gao, D. Li e H.-Y. Shum, «The Design and Implementation of XiaoIce, an Empathetic Social Chatbot,» *Comput. Linguistics,* vol. 46, p. 53–93, 2020.

[5] Z. Lin, P. Xu, G. I. Winata, F. B. Siddique, Z. Liu, J. Shin e P. Fung, «CAiRE: An End-to-End Empathetic Chatbot,» in *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, 2020.

[6] M. Asada, «Towards Artificial Empathy - How Can Artificial Empathy Follow the Developmental Pathway of Natural Empathy?,» *Int. J. Soc. Robotics,* vol. 7, p. 19–33, 2015.

[7] Ö. N. Yalçin, «Empathy framework for embodied conversational agents,» *Cogn. Syst. Res.,* vol. 59, p. 123–132, 2020.

[8] J. Gao, M. Galley e L. Li, «Neural Approaches to Conversational AI,» in *Proceedings of ACL 2018, Melbourne, Australia, July 15-20, 2018, Tutorial Abstracts*, 2018.

[9] M. Huang, X. Zhu e J. Gao, «Challenges in Building Intelligent Open-domain Dialog Systems,» *ACM Trans. Inf. Syst.,* vol. 38, p. 21:1–21:32, 2020.

[10] X. Gao, Y. Zhang, M. Galley, C. Brockett e B. Dolan, «Dialogue Response Ranking Training with Large-Scale Human Feedback Data,» in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, 2020.

[11] Y. Zhang, S. Sun, M. Galley, Y.-C. Chen, C. Brockett, X. Gao, J. Gao, J. Liu e B. Dolan, «DIALOGPT : Large-Scale Generative Pre-training for Conversational Response Generation,» in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations, ACL 2020, Online, July 5-10, 2020*, 2020.

[12] D. Adiwardana, M.-T. Luong, D. R. So, J. Hall, N. Fiedel, R. Thoppilan, Z. Yang, A. Kulshreshtha, G. Nemade, Y. Lu e Q. V. Le, «Towards a Human-like Open-Domain Chatbot,» *CoRR,* vol. abs/2001.09977, 2020.

[13] S. Roller, E. Dinan, N. Goyal, D. Ju, M. Williamson, Y. Liu, J. Xu, M. Ott, E. M. Smith, Y.-L. Boureau e J. Weston, «Recipes for Building an Open-Domain Chatbot,» in *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, EACL 2021, Online, April 19 - 23, 2021*, 2021.

[14] A. Ritter, C. Cherry e W. B. Dolan, «Data-Driven Response Generation in Social Media,» in *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, EMNLP 2011, 27-31 July 2011, John McIntyre Conference Centre, Edinburgh, UK, A meeting of SIGDAT, a Special Interest Group of the ACL*, 2011.

[15] O. Vinyals e Q. V. Le, «A Neural Conversational Model,» *CoRR,* vol. abs/1506.05869, 2015.

[16] A. Sordoni, M. Galley, M. Auli, C. Brockett, Y. Ji, M. Mitchell, J.-Y. Nie, J. Gao e B. Dolan, «A Neural Network Approach to Context-Sensitive Generation of Conversational Responses,» in *NAACL HLT 2015, The 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Denver, Colorado, USA, May 31 - June 5, 2015*, 2015.

[17] I. V. Serban, A. Sordoni, Y. Bengio, A. C. Courville e J. Pineau, «Building End-To-End Dialogue Systems Using Generative Hierarchical Neural Network Models,» in *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA*, 2016.

[18] I. V. Serban, A. Sordoni, R. Lowe, L. Charlin, J. Pineau, A. C. Courville e Y. Bengio, «A Hierarchical Latent Variable Encoder-Decoder Model for Generating Dialogues,» in *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA*, 2017.

[19] C. Sankar e S. Ravi, «Deep Reinforcement Learning For Modeling Chit-Chat Dialog With Discrete Attributes,» in *Proceedings of the 20th Annual SIGdial Meeting on Discourse and Dialogue, SIGdial 2019, Stockholm, Sweden, September 11-13, 2019*, 2019.

[20] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C.

Xu, T. L. Scao, S. Gugger, M. Drame, Q. Lhoest e A. M. Rush, «Transformers: State-of-the-Art Natural Language Processing,» in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, EMNLP 2020 - Demos, Online, November 16-20, 2020*, 2020.

[21] S. Bao, H. He, F. Wang, H. Wu, H. Wang, W. Wu, Z. Guo, Z. Liu e X. Xu, «PLATO-2: Towards Building an Open-Domain Chatbot via Curriculum Learning,» in *Findings of the Association for Computational Linguistics: ACL/IJCNLP 2021, Online Event, August 1-6, 2021*, 2021.

[22] S. Bao, H. He, F. Wang, H. Wu e H. Wang, «PLATO: Pre-trained Dialogue Generation Model with Discrete Latent Variable,» in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, 2020.

[23] J. Li, W. Monroe, A. Ritter, D. Jurafsky, M. Galley e J. Gao, «Deep Reinforcement Learning for Dialogue Generation,» in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, 2016.

[24] I. V. Serban, C. Sankar, M. Germain, S. Zhang, Z. Lin, S. Subramanian, T. Kim, M. Pieper, S. Chandar, N. R. Ke, S. Mudumba, A. de Brébisson, J. Sotelo, D. Suhubdy, V. Michalski, A. Nguyen, J. Pineau e Y. Bengio, «A Deep Reinforcement Learning Chatbot,» *CoRR,* vol. abs/1709.02349, 2017.

[25] A. Saleh, N. Jaques, A. Ghandeharioun, J. H. Shen e R. W. Picard, «Hierarchical Reinforcement Learning for Open-Domain Dialog,» in *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, 2020.

[26] R. Lowe, M. Noseworthy, I. V. Serban, N. Angelard-Gontier, Y. Bengio e J. Pineau, «Towards an Automatic Turing Test: Learning to Evaluate Dialogue Responses,» in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, 2017.

[27] M. Brambilla, A. J. Sabet e A. E. Sulistiawati, «Conversation Graphs in Online Social Media,» in *Web Engineering - 21st International Conference, ICWE 2021, Biarritz, France, May 18-21, 2021, Proceedings*, 2021.

[28] Z. Lin, A. Madotto, J. Shin, P. Xu e P. Fung, «MoEL: Mixture of Empathetic Listeners,» in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, 2019.

[29] R. Zandie e M. H. Mahoor, «EmpTransfo: A Multi-Head Transformer Architecture for Creating Empathetic Dialog Systems,» in *Proceedings of the Thirty-Third International Florida Artificial Intelligence Research Society Conference, Originally to be held in North Miami Beach, Florida, USA, May 17-20, 2020*, 2020.

[30] N. Lubis, S. Sakti, K. Yoshino e S. Nakamura, «Eliciting Positive Emotion through Affect-Sensitive Dialogue Response Generation: A Neural Network Approach,» in *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, 2018.

[31] N. Lubis, S. Sakti, K. Yoshino e S. Nakamura, «Unsupervised Counselor Dialogue Clustering for Positive Emotion Elicitation in Neural Dialogue System,» in *Proceedings of the 19th Annual SIGdial Meeting on Discourse and Dialogue, Melbourne, Australia, July 12-14, 2018*, 2018.

[32] J. Shin, P. Xu, A. Madotto e P. Fung, «Generating Empathetic Responses by Looking Ahead the User's Sentiment,» in *2020 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2020, Barcelona, Spain, May 4-8, 2020*, 2020.

[33] R. W. Picard, «Affective Computing for HCI,» in *Human-Computer Interaction: Ergonomics and User Interfaces, Proceedings of HCI International '99 (the 8th International Conference on Human-Computer Interaction), Munich, Germany, August 22-26, 1999, Volume 1*, 1999.

[34] V. Scotti, R. Tedesco e L. Sbattella, «A Modular Data-Driven Architecture for Empathetic Conversational Agents,» in *IEEE International Conference on Big Data and Smart Computing, BigComp 2021, Jeju Island, South Korea, January 17-20, 2021*, 2021.

[35] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei e I. Sutskever, «Language models are unsupervised multitask learners,» *OpenAI blog,* vol. 1, p. 9, 2019.

[36] D. P. Kingma, T. Salimans, R. Józefowicz, X. Chen, I. Sutskever e M. Welling, «Improving Variational Autoencoders with Inverse Autoregressive Flow,» in *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, 2016.

[37] S. Park e J. Lee, «Finetuning Pretrained Transformers into Variational Autoencoders,» *CoRR,* vol. abs/2108.02446, 2021.

[38] E. Jang, S. Gu e B. Poole, «Categorical Reparameterization with Gumbel-Softmax,» in *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*, 2017.

[39] R. J. Williams, «Simple Statistical Gradient-Following Algorithms for Connectionist Reinforcement Learning,» *Mach. Learn.,* vol. 8, p. 229–256, 1992.

[40] I. Loshchilov e F. Hutter, «Decoupled Weight Decay Regularization,» in *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*, 2019.

[41] Y. Li, H. Su, X. Shen, W. Li, Z. Cao e S. Niu, «DailyDialog: A Manually Labelled Multi-turn Dialogue Dataset,» in *Proceedings of the Eighth International Joint Conference on Natural Language Processing, IJCNLP 2017, Taipei, Taiwan, November 27 - December 1, 2017 - Volume 1: Long Papers*, 2017.

[42] H. Rashkin, E. M. Smith, M. Li e Y.-L. Boureau, «Towards Empathetic Open-domain Conversation Models: A New Benchmark and Dataset,» in *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, 2019.

[43] S. Zhang, E. Dinan, J. Urbanek, A. Szlam, D. Kiela e J. Weston, «Personalizing Dialogue Agents: I have a dog, do you have pets too?,» in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, 2018.

[44] E. Dinan, S. Roller, K. Shuster, A. Fan, M. Auli e J. Weston, «Wizard of Wikipedia: Knowledge-Powered Conversational Agents,» in *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*, 2019.