Sentiment-Aware Conversational Agent

***Abstract*—Current state-of-the-art dialogue text generation models rely on large amounts of data in order to implicitly learn how to generate fluent and appropriate text. Some applications, such as customer support, have started to rely on such systems to retain and increase the confidence of customers with a fast and effective resolution of possible problems. However, the data available for such applications is often scarce, which might not allow to properly train these models, leading to automatic generic answers, which is problematic, since sentiment is often regarded as an important aspect of customer satisfaction.**

We propose to tackle these issues by developing an end-to-end sentiment-aware conversational agent. To do so, we will develop three models: a sentiment classification model, tasked with classifying the sentences of the dialogue; a reply sentiment prediction model, which leverages the context of the dialogue in order to predict an appropriate sentiment for the agent to express in its reply; and a text generation model, which is conditioned on the predicted sentiment and the context of the dialogue, in order to produce a reply that is both context and sentiment appropriate.

Both automatic metrics and human evaluation show that explicitly guiding the text generation model with a pre-defined set of sentences leads to clear improvements, in particular for models fine-tuned with small datasets. Finally, we show that the reply sentiment prediction model is the bottleneck of the system, and discuss future approaches.

***Index Terms*—Natural Language Processing; Sentiment Classification; Reply Sentiment Prediction; Conditioned Text Generation; Sentiment-Aware Conversational Agent.**

I. Introduction

CONVERSATIONAL agents have become popular over the years in various forms, such as personal assistants, or as an automatic way of a company to provide information to its customers, among others. With the emergence of social networks, which made dialogue data become more available, and the increase of computational power, machine learning and, in particular, deep learning approaches, were able to improve the performance of dialogue conversational agents through the use of neural network models and word embeddings [1]–[6], which require high amounts of data in order to produce grammatically correct and adequate answers. In particular, [7] shows that training these models in a large empathetic corpus enables them to generate text that expresses emotions. However, [8] demonstrates that it is hard to achieve good generation results when fine-tuning current models with small datasets. Furthermore, in [7] it is shown that models trained on "spontaneous internet conversation data are not rated as very empathetic". The problem is that for specific-domains, such as customer support, the amount of data is usually scarce or not publicly available due to privacy constraints. Additionally, emotion-labelled corpora, in particular with a more fine-grained set of labels, is difficult to build in a large scale. Existing works have started to find ways to explicitly condition text generation models in order to produce certain

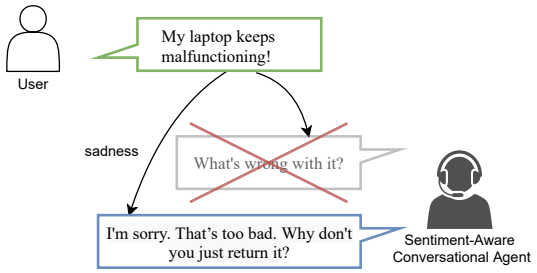

Fig. 1. Example of a conversation when using the sadness emotion is appropriate.

attributes, such as emotion [9], [10], personality traits [11], among others [12], [13]. The drawback of these approaches is that they can not be used in a dialogue setting, given that they do not introduce a mechanism to automatically predict the next appropriate attribute, as discussed in [14]. Considering once again the customer support scenario, nowadays, companies started to rely on conversational agents, which raises a concern: receiving what seems like an automated and generic reply message might not please most customers, as mentioned in [15]. However, available conversational agents for customer support are able to produce answers that are both grammatically correct and useful, but they do not take into account the emotions expressed by the customer [16]. Thus, an end-to-end system that aids customer support by combining informative answers with the appropriate reply emotion that best suits the customer's state-of-mind, could help improve customer satisfaction by providing a fast, adequate and non-generic answer. For example, in Figure 1, we can observe that, while both answers can be considered as correct, an answer that expresses an appropriate emotion may be more satisfying for the user, rather than the generic reply.

Following [17], which highlights that in human-human conversations the emotions expressed in two subsequent utterances from different speakers often change and therefore are important when predicting the "correct emotion for an upcoming response before generation", in this work we propose an end-to-end sentiment-aware conversational agent, which can be observed in Figure 2.

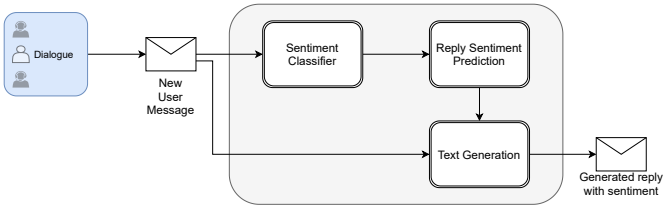

Fig. 2. Proposed end-to-end sentiment-aware conversational agent.

The goal of the proposed conversational agent is to be able to generate sentiment-appropriate sentences in real-time through the use of three models:

- A Sentiment Classification model, which classifies the user's input sentence;
- A Reply Sentiment Prediction model, which predicts the appropriate reply sentiment that should be expressed by the conversational agent;
- A Text Generation model, which generates a sentence that is context-aware and expresses the predicted sentiment.

The main contributions of this work are the following:

1) We explore an end-to-end conversational-agent approach which is able to: leverage information about the sentiment of the received input sentence; consider the most appropriate reply sentiment to that input; and generate text conditioned on a given sentiment;
2) We explore the possibility of including multiple context sentences, and the use of retrieval augmentation techniques in order to bias the model towards the correct sentiment label in the sentiment classification and reply sentiment prediction models;
3) We adapt the work proposed by [11] to the task of sentiment-conditioned text generation. In particular, we show that this adaptation has clear benefits for models fine-tuned with small datasets;
4) We conduct a human evaluation that aimed to assess the adequacy and sentiment accuracy of the proposed system. We show that this evaluation correlated well with the used automatic metrics, which motivates their use during the development of new models.

The overview of the present document is described as follows: in Section II we describe some state-of-the-art models used for the tasks of sentiment classification, reply sentiment prediction and text generation; in Section III we present the models that are going to be used in order to build the sentiment-aware conversational agent. Section IV, consists on an experimental cycle performed for each model, along with support experiments and examples that help validate the obtained results. After obtaining the automatic metrics' results, we perform a human evaluation, in order to correlate the results obtained with the automatic metrics; we finalize this document with a conclusion of our findings in Section V, along with future considerations for this work.

II. Related Work

In this section we will describe the current most relevant state-of-the-art regarding sentiment classification, prediction and conditioned text generation.

A. Sentiment Classification

The current state-of-the-art for text classification tasks, and particularly sentiment classification, is making use of dense contextual word embedding models, such as BERT [18] or RoBERTa [19], via transfer learning, i.e., using the embeddings obtained from a pre-trained model to train a classification model. The motivation for using pre-trained models is that the embeddings learned by larger models and datasets can encode meaningful features for the task at hand that we would not be able to obtain by learning the embeddings from scratch. As described in [18] and [20], fine-tuning such models for the task of sentiment classification involves using the final hidden state of the model corresponding to the first input token (the `[CLS]` token, as the sentence representation, $o_{\texttt{[CLS]}}$. Then, an extra linear layer with the softmax transformation is added to the dense contextual word embedding model to predict the probability of the sentence belonging to label l,

$$p(l|o_{\texttt{[CLS]}}) = \text{Softmax}(W \cdot o_{\texttt{[CLS]}}), \tag{1}$$

where W represents the learned weights by the classification model. Other works on sentiment classification made use of contextual, speakers, speech acts and topics information [21], further pre-trained BERT with a dataset similar to the target sentiment-labelled dataset [22], or made use of acyclic graph networks to "model the information flow between long-distance conversation background and nearby context" [23].

B. Sentiment Prediction

One critical aspect of systems that deal with sentiment-aware text generation in a dialogue context is how to define the appropriate sentiment for the upcoming reply. A possible way to do so is to automatically learn the sentiment that should be expressed through the use of sentiment prediction models. [24] makes use of a Support Vector Machine model [25] in order to predict the sentiment of an upcoming agent sentence in a customer support scenario. In particular, dialogue and textual features, such as the time between interactions, or the emotion of previous sentences, are used as additional information to aid the model. In [26] an LSTM is used to classify the sentiment of the next utterance. To do so, the network is trained on a dataset where each example corresponds to two consecutive utterances from two different speakers (in a dialogue style) and the target is the sentiment of the sentence that follows. [17] explores a similar approach, using a bi-LSTM as the classification model, but instead of using a pair of utterances, it uses multiple examples of possible answers to the first utterance in order to capture the next reply emotion. [27] aims to simulate the emotion transition of humans in a dialogue. To do so, it makes use of the Valence-Arousal-Dominance emotion space [28], which encodes the emotion of words in a 3-dimensional vector space, to calculate the "emotion transition as the variation between the preceding emotion and the response emotion". It is important to notice, that some approaches described in this section use dataset-specific characteristics that are not available in all corpora. Furthermore, it is interesting to observe that the reported scores for the aforementioned approaches fall short of expectations, which shows how difficult it is to predict the correct sentiment of the next sentence with current models.

C. Text Generation

We will consider two different alternatives of conditioning text with a given attribute: controllable text generation and dialogue text generation. The key difference between the two is that controllable text generation involves pre-defining an attribute that is used as an extra input to condition the

generated text, while dialogue text generation consists of an end-to-end system that is able to implicitly learn from the data how to generate the desired properties. In particular, since controllable text generation models depend on the conditioning attribute to defined explicitly, they require a mechanism to predict the attribute of the upcoming sentence in order to be used as an end-to-end dialogue system.

1) Controllable Text Generation: In [9], "the problem of generating emotional responses in open-domain conversational systems" is addressed through the use emotion embeddings based on an external vocabulary memory, which are used to condition a sequence-to-sequence (Seq2Seq) [29] model. During a dialogue, this model is only able to express a single emotion. [16] proposed a "Tone-aware Chatbot for Customer Care on Social Media" based on a Seq2Seq architecture with LSTMs. It concatenates an extra tone vector to the input word vector in each step of the decoder with information related to three tones: empathetic, passionate and neutral. These tones are the ones the authors found to be the most significant for customer support satisfaction and therefore are used when responding to customers. [30] builds a collection of dialogues from Twitter that include emojis and assumes the emojis as the underlying emotion in the sentence. Then, it trains a Seq2Seq model on this corpus, and conditions the decoding with an embedding corresponding to the target emoji. [31] improves [9] by, besides using emotion embeddings to condition a Seq2Seq model, also penalizing neutral words and forcing the model to generate words related with the desired emotion. [32] proposes a similar idea, also using an emotion embedding to condition the language model with a target sentiment, but using a Transformer model [5], in particular, the GPT-2 language model [33].

2) Dialogue Text Generation: The state-of-the-art literature in dialogue text generation mainly consists in data-driven end-to-end models which are capable of generating fluent, appropriate, and meaningful responses in a dialogue setting, by using previous context sentences as input to text generation models. One of the first successful approaches was proposed in [3], which leverages the Seq2Seq architecture and recurrent neural networks to predict an upcoming sentence by using as input to the model the previous context of the conversation, thus, allowing the model to be used in a dialogue scenario. [11] applies this idea to the Transformer architecture, and fine-tunes the GPT-2 model using two additional special tokens that are used to separate the sentences belonging to different speakers in the model's input. Regarding implicit text generation with sentiment, [7] fine-tunes a GPT-2 model with the EmpatheticDialogues dataset, proposed in the same work, and concludes that a large-scale empathetic corpus enables the models to express appropriate emotions in dialogue.

3) Controllable Dialogue Text Generation: Controllable dialogue text generation approaches make use of a mechanism that is able to condition the text generation model automatically, thus allowing the system to work end-to-end. In this section we will consider works that have done this in two ways: first, using a pre-defined set of rules or heuristics; second, through the use of data-driven language models.

[34] proposed a conversational model which embeds words using the Valence-Arousal-Dominance (VAD) emotion space [28], and explores decoding by using different Beam Search techniques that aim to incorporate affective diversity in candidate outputs. Furthermore, it designs "training loss functions to explicitly train an affect-aware Seq2Seq conversation model", following three heuristics: minimizing affective dissonance (the generated text emotion should be similar to the input's emotion); maximizing affective dissonance (the generated text emotion should not be aligned with the input's emotion); and maximizing affective content (the generated text emotion should avoid being neutral). [10] adopts an "emotion mining from text" classifier, developed in [35], to classify the emotions expressed in previous conversation context, and uses this information, together with pre-defined mapping rules defined by the authors, to decide which emotion should be expressed in the reply. This emotion is then either concatenated to the input of the model, or injected into the decoder. [36] uses a neural network to predict which emotion keyword, selected from a pre-defined lexicon dictionary [37], should be used in the response, which, similarly to [10], is then introduced in the decoder. In [38], the VAD emotion space is used to understand the emotion expressed in previous context. The model's response is then conditioned by following a similar or opposite emotion to the speaker's assessed emotion.

[14] argues that approaches that rely on a pre-defined set of rules or heuristics, such as the previously mentioned, are not supported by psychology literature, and therefore emotional interactions in human-human conversations should be explored instead with a large-scale emotional corpus by using data-driven language models. Some works leverage a multi-task approach that jointly trains an emotion encoder and the text generation model, which is conditioned on the emotional state assessed by the emotion encoder [39], [40]. On the other hand, [14] incorporates a emotion/intent predictor, which is separately trained from the text generation model, with the goal of deciding the emotion/intent for the reply to be generated. That emotion is predicted based on previous context, and is then encoded and fed to the text generation model. An interesting aspect to notice about the aforementioned works is that the evaluation done focuses on whether the generated texts are empathetic, and not whether they are generating a specific emotion. During our work we will also be evaluating whether the developed models are capable of generating specific sentiments.

III. Sentiment Aware Conversational Agent

The goal of this work is to create a conversational system that is sentiment aware. In order to achieve that, it is not only important to understand the sentiment behind each sentence in the dialogue, but also the appropriate reply sentiment in the context of the conversation, in order to condition the text generation model. To do so, we propose three modules: a sentiment classification model, responsible for classifying the sentences with a sentiment; a reply sentiment prediction model, which is able to predict the appropriate reply sentiment given the conversation's context; and a text generation model, which is conditioned on the predicted sentiment and on the past conversation context.

A. Sentiment Classification and Reply Sentiment Prediction

Given that both sentiment classification and reply sentiment prediction are classification tasks, in order to develop them we are going to make use of pre-trained Transformer models, such as BERT [18]. We will improve upon this base model, by using previous dialogue context, and the sentiment labels of similar examples retrieved from the train corpus as input to the model. At the end of this section, we will describe the reply sentiment prediction classifier and how, given a dialogue context, it can be used to predict the appropriate sentiment that should be conveyed by the reply of the conversational system.

1) Contextual Sentiment Classification: In order to add context to the input of the model, we will take advantage of a particularity of the BERT architecture, the `[SEP]` token. Since both BERT and RoBERTa models are limited to 512 input tokens, similarly to [22], we consider that the first sentence after the `[CLS]` token is the sentence we are trying to classify, and it is followed by its context. Given a dialogue $D = (s_1, s_2, ..., s_n)$, with n equal to the number of sentences in the dialogue, in order to classify the sentence s_i with x sentences as context, the input to the model is $\text{concat}(s_i, s_{i-1}, ..., s_{i-x})$.

2) Sentiment Classification using Retrieval Augmentation: Due to the recent success of retrieval augmentation approaches in NLP tasks [41]–[43], including sentiment classification [44], we explore a mechanism that relies on nearest neighbors. In particular, the work of [44] uses this logic for models that are based on the LSTM model. Since we are dealing with Transformer models, we will adapt their approach, with a focus on how to provide the retrieved labels as input.

The first step involves for each train/development/test example to find the nearest training example. To do so, we use the Sentence Transformer [45][1] library to create sentence representations of all examples using the `paraphrase-distilroberta-base-v1` model. Next, we make use of the FAISS [46][2] library to build an index with the sentence embeddings that belong to the train set, and also to find the closest training examples. In particular, we chose the Euclidean distance to calculate the distances between the examples. Finally, each train/development/test example is assigned a label corresponding to the label of the closest training example. The goal is to incorporate this information into the Transformer model in order to guide it towards the corresponding retrieved label.

After retrieving the nearest neighbors information, we apply it to the Transformer, which is where our method differs from [44]. In order to do so, we first initialize an extra set of embeddings, which we will call Sentiment Embeddings (SE), one for each sentiment label, which are trained along with the model. Then, for each example, we incorporate the nearest training example label in the Transformer model, by concatenating to the output of the Transformer model, after pooling, the embedding corresponding to the label of the nearest training example.

We were inspired by the initialization of the memory state of the LSTM described in [44], where different initialization approaches were experimented with. Since the Transformer model does not have a memory state, we initialize instead the embedding of each sentiment with the average of the sentences' corresponding to that given sentence. E.g., the sentiment embedding for *joy* will be initialized with the average of the embeddings corresponding to the sentences labelled with the sentiment *joy* in our training set. For this use case, we use the average of the token embeddings of the last hidden layer as the embedding of a given sentence. Other experiments were done, in particular, we experimented with incorporating the sentiment embeddings by adding them to the input embeddings of the model, but it resulted in a poor classification performance.

3) Reply Sentiment Prediction: This model receives as input the previous context of a dialogue in order to predict the appropriate sentiment to be expressed in the conversational agent's next reply. This aspect makes it crucial for the usage of the system, given that it will be the model that will allow the conversational agent to be sentiment-aware.

Similarly to the contextual sentiment classification, we will also make use of the `[SEP]` token to separate the different sentences that are part of the input. However, there are two main differences: first, the input of the reply sentiment prediction model corresponds only the previous context sentences; secondly, the gold label is the sentiment of the upcoming sentence. In Table I we can observe an example of the input of the model. The last sentence in a dialogue is not considered as a valid example, given that there are no sentences that follow it. In this task we can also use an arbitrary number of sentences as context. We will use as default setup the last two previous subsequent sentences.

In addition to the retrieval augmentation, which will be used in a similar manner for this task, we can also add to the input of the model information about the sentiment of the sentences from the context.

B. Text Generation

During a dialogue, the text generation model will receive the sentiment predicted by the reply sentiment prediction model, as well as the previous context of the conversation, to generate a sentence that is not only appropriate given a context, but that expresses the predicted sentiment. To do so, we are going to adapt part of the work by [11], that proposes a model that is able to leverage a set of sentences that describe a given persona in order to generate text that is coherent with the persona. We will adapt this concept and develop a sentiment lexicon knowledge base, which will be used to make the conversational agent sentiment-aware.

In Figure 3, we can observe the input and output of the GPT-2 model. In order to make a language model suitable for the dialogue task, as described in [11], we concatenate previous context of the dialogue (history) to the input of the model. This is possible by introducing two extra special tokens to the model, `<speaker1>`, which indicates the beginning of a sequence from the user, and `<speaker2>`, which indicates the beginning of a sequence from the bot, in addition to the existing special tokens `<BOS>`, which indicates

[1]https://www.sbert.net
[2]https://faiss.ai

Sentence	Label	Representation	Label
Does it cost anything?	NEU	`[CLS]Does it cost anything?[SEP]`	NEU
Yeah 20$ per month.	NEU	`[CLS]Yeah 20$ per month.[SEP]Does it cost anything?[SEP]`	SUR
Ohh!	SUR	-	-

TABLE I

EXAMPLE OF A DIALOGUE FROM THE EMOTIONPUSH DATASET AND THE SENTENCE REPRESENTATION AND CORRESPONDING LABEL FOR THE RESPONSE SENTIMENT PREDICTION TASK, WITH TWO PREVIOUS SENTENCES AS CONTEXT ($x = 2$). NEU CORRESPONDS TO THE LABEL *Neutral*, AND SUR TO THE LABEL *Surprise*.

the beginning of a sequence, and <EOS>, which indicates the end of a sequence. Furthermore, in this figure we can also see the autoregressive property of this model. Autoregressive language models use the left-wise context of a sentence in order to calculate the probability of the next word. At each timestep t, the input of the model is $\text{concat}(h; w_{0:t-1})$, with h representing the history, and $w_{0:t-1}$ representing a sequence of generated words from timestep 0 until $t-1$. The output is w_t, which is the word generated at timestep t.

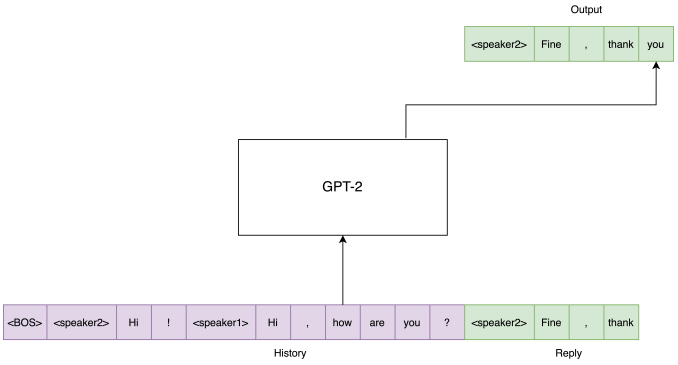

Fig. 3. Example of the input and output of the base GPT-2 model adapted for the dialogue task.

In order to condition the text generation base model with a desired sentiment, we follow the work of [11]. Instead of a persona, we will input lexicon that represents the desired sentiment, by concatenating the lexicon to the beginning of the input of the model. An advantage of this model is that we can experiment with various sentiment lexicons, which can consist of a single word for each sentiment (for example, the name of the sentiment), expressions, or full sentences. For instance, by adding the sentiment lexicon of the *anger* emotion, we expect the model to produce a more sentiment appropriate sentence ("I am annoyed"), than the base model ("Fine, thank you"), given the same history.

IV. EXPERIMENTS

In this section we will present the results obtained for each of the proposed models applied to the EmotionPush [47] and DailyDialog [48]. The EmotionPush corpus is composed of 1000 private conversations from Facebook Messenger. The DailyDialog corpus was built from websites that are used to practice English dialogue in daily life scenarios. It contains 13118 multi-turn dialogues, divided in 10 themes such as: Finance, Politics, Health, Work, etc. Both corpora use as labels the six Ekman's basic emotions [49], and neutral. We highlight the fact that both corpora are highly unbalanced, with the *neutral* label composing over 80% of the examples.

The models were implemented using PyTorch Lightning [50] and the HuggingFace Transformers [51] library. In particular, we modified the code bases of the HLT-MAIA Emotion-Transformer repository[3] for the sentiment classifier and reply sentiment prediction models, and the lightning-convai repository[4] for the text generation model.

The sentiment classifier and reply sentiment prediction models were trained for a maximum of 40 epochs, using the cross entropy loss, with four validation steps per epoch, stopping the training after 10 consecutive validation steps without improvement. The checkpoint used to evaluate the model was the one that achieved the highest validation macro-F1 value. We follow the work by [52] and use the Adam optimizer [53] with a discriminative learning rate of 1×10^{-3}, except for the Transformer model that has a learning rate of 5×10^{-6}. For the Transformer model we apply a layer-wise learning rate decay of 0.95 after each training step. We apply a dropout [54] of 0.4 to the sentence embeddings during training. We use a real batch size of 16 whenever the GPU's memory allowed it, but we used gradient accumulation to simulate a batch size of 32.

The text generation models were trained for a maximum of 40 epochs, using the negative log-likelihood loss, with four validation steps per epoch, stopping the training after 12 consecutive validation steps without improvement. We use the Adam optimizer [53] with a learning rate of 5×10^{-6}. The checkpoint used to evaluate the model was the one that achieved the lowest validation negative log-likelihood loss value. Due to computational constraints, we always use the two most recent context sentences from the dialogue as input to the model. We use a real batch size of 4 whenever the GPU's memory allowed it, but we used gradient accumulation to simulate a batch size of 16.

All other hyperparameters were kept as default.

Due to the poor balancing of the datasets used, we will evaluate the sentiment classifier and the reply sentiment prediction using the micro (*m*) and macro (*M*) averages. The macro-F1 evaluates whether the model is able to classify examples of all labels equally, while the micro-F1 evaluates the number of examples classified correctly. Furthermore, given the distribution of examples labelled as *neutral* (the majority class) vs. all other labels, we will also evaluate our models both with the majority class and without the majority class (micro/Macro-No Majority Class metric which we will refer to as *m/M-NMC*). This will allow us to have a better understanding of how the models are performing on less represented sentiments.

Regarding the text generation model, our choice of automatic evaluation metrics aims to measure if the model is

[3]https://github.com/HLT-MAIA/Emotion-Transformer
[4]https://github.com/HLT-MAIA/lightning-convai

able to generate a sentence that expresses a desired sentiment without compromising the quality of the text. Therefore, first, regarding the quality of the generated text, we will focus on two metrics: **Perplexity (PPL)**, which is a metric used to compare language models. The model with the lowest PPL has a higher probability of correctly generating an unseen example from a test set; and the **Sentence Embedding Similarity (SES)**, which calculates the cosine similarity between the embeddings of the generated and the gold examples. Similarly to [14], we use the Sentence Transformer [45] library to create sentence representations of both sentences, using the `paraphrase-distilroberta-base-v1` model. Then, we calculate the cosine similarity between the two representations. The SES is the average of the cosine similarities obtained for all examples in the test set. Second, to evaluate if the generated text is expressing the appropriate sentiment, we will use the sentiment classification model. This model will classify the generated sentences and evaluate them using the aforementioned F1 metrics.

A. Sentiment Classification

Using the development set, we were able to find the following optimal setup: a RoBERTa-large model, that receives as input the concatenation of the sentence to be classified, with the last previous context sentence; a linear classification layer that receives as input the concatenation of the `[CLS]` token embedding of the last 4 hidden layers (*concat4* pooling); and the retrieval augmentation method previously described. We use as a baseline the RoBERTa-large, as described in [19].

We will start by comparing the performance of our model (SA Model) with the baseline, for both development and test sets on the EmotionPush corpus. The results can be seen in Table II. It can be observed that our model improves all metrics, except the micro-F1 on the test set where it maintains the same value. More notably, it is able to improve the macro-F1 metric by 5.3 points on the development set, and 8.7 points on the test set. These improvements are also noticeable on the M-NMC metric, where our model improves 6.1 points on the development set and 10 points on the test set.

		F1			
		m	**m-NMC**	**M**	**M-NMC**
Dev	Baseline	76.8	51.9	47.5	41.8
	SA Model	**77.0**	**53.6**	**52.8**	**47.9**
Test	Baseline	**78.9**	57.6	45.4	39.2
	SA Model	**78.9**	**58.2**	**54.1**	**49.2**

TABLE II
COMPARISON BETWEEN THE BASELINE AND OUR BEST SENTIMENT ANALYSIS MODEL ON THE DEVELOPMENT AND TEST SETS OF THE EMOTIONPUSH CORPUS.

In order to further validate our results we perform an ablation study using the test set, with four experiments defined as follows: **+RoBERTa-base**, where we replace RoBERTa-large by RoBERTa-base; **+CLS**, where we replace the *concat4* pooling by the embedding of the `[CLS]` token of the last hidden layer; **-Context 1**, where we no longer use context in the input of our model; **-Ret. Aug.**, where we remove the retrieval augmentation from the model.

	F1			
	m	**m-NMC**	**M**	**M-NMC**
SA Model	**78.9**	58.2	54.1	49.2
+ RoBERTa base	-0.7	-0.8	-4.2	-4.7
+ CLS	-0.9	-0.9	-1.6	-1.7
- Context 1	-2	-2.6	-8.8	-9.9
- Ret. Aug.	-0.7	**+0.3**	**+3.1**	**+3.6**

TABLE III
ABLATION STUDY ON THE EMOTIONPUSH TEST SET.

The results obtained on the ablation study can be observed in Table III. Removing the context is what impacts the model the most, which tells us it was the most significant addition to our model. Additionally, replacing the RoBERTa-large by the RoBERTa-base and the *concat4* by the CLS pooling option, also worsens all metrics. Interestingly, the retrieval augmentation worsened our results on the test set.

Regarding the results on the DailyDialog corpus, since on the ablation study performed on the EmotionPush corpus we found that removing context and retrieval augmentation had the most impact on the model, we will only focus on those changes for this corpus both on the development and test sets.

A summary of the results obtained on the development set can be seen in Table IV. First, we can start by noting that our model does not improve the baseline on most metrics, except the micro-F1 metric that improves 0.3 points. Secondly, we can observe that removing the context or retrieval augmentation, has a minimal impact on the metrics.

	F1			
Model	**m**	**m-NMC**	**M**	**M-NMC**
Baseline	89.3	59.4	47.5	39.8
SA Model	**89.6**	59.3	47.4	39.5
- Context 1	-0.3	0	0	+0.1
- Ret. Aug.	0	**+0.2**	**+0.3**	**+0.4**

TABLE IV
RESULTS OBTAINED ON THE DAILYDIALOG DEVELOPMENT SET. THE RESULTS ON THE ABLATION STUDY ARE A COMPARISON WITH THE RESULTS OBTAINED FOR THE SA MODEL.

In Table V, we have the results on the test set. Contrarily to the results obtained on the development set, all metrics improve significantly when compared to the baseline. The micro-F1 metric improves by 0.5 points, and without the majority class by 1.6 points. Regarding the macro-F1 metric, it improves by 2.9 points and without the majority class by 3.4 points. Interestingly, on the test set both removing the context and retrieval augmentation worsen the results, which tells us that both methods are helping in the classification.

	F1			
Model	**m**	**m-NMC**	**M**	**M-NMC**
Baseline	84.5	56.5	48.1	41.0
SA Model	**85.0**	**58.1**	**51.0**	**44.4**
- Context 1	-0.6	-1.6	-2.5	-3.0
- Ret. Aug.	-0.6	-1.3	-1.0	-1.2

TABLE V
RESULTS OBTAINED ON THE DAILYDIALOG TEST SET. THE RESULTS ON THE ABLATION STUDY ARE A COMPARISON WITH THE RESULTS OBTAINED FOR THE SA MODEL.

B. Reply Sentiment Prediction

Using the development set, we were able to find the following optimal setup: a RoBERTa-large model, that receives as input a concatenation of the last four context sentences; a linear classification layer that receives as input the concatenation of the `[CLS]` token embedding of the last 4 hidden layers (*concat4* pooling). In particular, incorporating the labels of the context sentences to the input of the model did not improve performance. For that matter, the sentiment classification model is not part of the conversational agent, and it is only used as a metric to evaluate whether the generated sentences express the desired sentiment. Additionally, the retrieval augmentation methods also did not improve performance. We use as a baseline the RoBERTa-large model that receives as input the last two context sentences.

		F1			
		m	**m-NMC**	**M**	**M-NMC**
Dev	Baseline	**69.0**	18.0	15.0	5.5
	RSP Model	66.5	**19.6**	**17.8**	**8.8**
Test	Baseline	**66.0**	14.4	13.3	3.8
	RSP Model	64.2	**21.1**	**15.7**	**6.7**

TABLE VI
COMPARISON BETWEEN THE BASELINE AND OUR BEST REPLY SENTIMENT PREDICTION MODEL ON THE DEVELOPMENT AND TEST SETS OF THE EMOTIONPUSH CORPUS.

In Table VI we can observe the comparison of our model versus the baseline in both the development and test sets of the EmotionPush corpus. In the development set we can see how the *micro*-F1 does not improve when compared to the baseline. Nevertheless, our model improves all other metrics (*m-NMC* by 1.6 points, *M* by 2.8 points, and *M-NMC* by 3.3 points), which shows how our model is better at generalizing for less represented sentiments. Regarding the performance on the test set, the conclusions are very similar. The *micro*-F1 is higher by 1.8 points on the baseline. Despite this our model also outperforms the baseline in all other metrics (*m-NMC* by 6.7 points, *M* by 2.4 points, and *M-NMC* by 2.9 points), which validates the improvements when compared to the baseline.

In order to further validate our results, we will perform an ablation study on the test set using a similar method to the one defined for sentiment classification.

	F1			
	m	**m-NMC**	**M**	**M-NMC**
RSP Model	64.2	**21.1**	15.7	**6.7**
+ RoBERTa base	**+3.4**	-6.8	**+2.5**	-3.1
+ CLS	+2.6	-4.9	-1.6	-2.1
+ Context 2	+0.2	-3.4	-1.3	-1.5
+ Ret. Aug.	+0.6	-3.2	-0.3	-0.3

TABLE VII
ABLATION STUDY ON THE EMOTIONPUSH TEST SET.

In Table VII, we can observe the ablation study done on the test set. We can start by observing how all models have a higher *micro*-F1 than the RSP model. The remaining major metrics all perform worse than the RSP model. These results show how our model is doing a trade-off between a lower F1 in the majority (*neutral*) label and a higher F1 on the less represented sentiments. Another important aspect is the performance of the model with retrieval augmentation, which does not improve results.

Regarding the results on the DailyDialog corpus, the ablation study performed on the EmotionPush showed that all introduced changes were impacting the final model. For that matter, in addition to showing the baseline and the best model results, we will also perform the same ablation study done in the previous section to the DailyDialog corpus, for both development and test sets.

	F1			
Model	**m**	**m-NMC**	**M**	**M-NMC**
Baseline	86.6	**40.9**	28.0	17.2
RSP Model	85.4	39.9	27.6	16.9
+ RoBERTa base	-0.6	-3	-3.7	-4.3
+ CLS	+0.8	-0.4	-1.4	-1.7
+ Context 2	-0.2	-2.5	-0.9	-1.1
+ Ret. Aug.	**+2**	-5.9	**+1.2**	**+1.5**

TABLE VIII
COMPARISON BETWEEN THE BASELINE AND RSP MODEL RESULTS OBTAINED, AND ABLATION STUDY PERFORMED ON THE DAILYDIALOG DEVELOPMENT SET. THE RESULTS ON THE ABLATION STUDY ARE A COMPARISON WITH THE RESULTS OBTAINED FOR THE RSP MODEL.

In Table VIII we can observe the results obtained on the development set for both the baseline and our model (RSP Model), as well as the ablation study. Firstly, we can observe that even using a larger number of training examples, the reply sentiment prediction task is still a hard task to perform well in. Comparing the baseline with the RSP Model, we can observe that this setup did not improve the base model on most metrics. Furthermore, if we consider the ablation study, we can see that none of the changes had a very strong impact on the performance either. Interestingly, on this corpus the retrieval augmentation improved some of the metrics.

	F1			
Model	**m**	**m-NMC**	**M**	**M-NMC**
Baseline	80.7	40.1	33.8	24.6
RSP Model	80.4	**42.8**	35.0	26.1
+ RoBERTa base	-1.2	-1.5	-2.9	-3.2
+ CLS	+0.7	-1.3	-0.4	-0.6
+ Context 2	-0.2	-1.7	**+0.5**	**+0.5**
+ Ret. Aug.	**+1.6**	-6.1	-1.9	-2.4

TABLE IX
COMPARISON BETWEEN THE BASELINE AND RSP MODEL RESULTS OBTAINED, AND ABLATION STUDY PERFORMED ON THE DAILYDIALOG TEST SET. THE RESULTS ON THE ABLATION STUDY ARE A COMPARISON WITH THE RESULTS OBTAINED FOR THE RSP MODEL.

The results obtained on the test set, which can be observed in Table IX, allow for a different set of conclusions. First, our introduced changes improve the baseline. Second, contrarily to the evaluation done on the development set, retrieval augmentation does not improve the results on most metrics.

C. Text Generation

The developed conditioned text generation model consists on a DialoGPT-small model, plus the concatenation of the desired sentiment's lexicon to the input of the model. Several lexicons were experimented with, such as using the sentiment's name, or retrieving the most common terms in the training set

for each sentiment, but the option that showed the best results was to use a pre-defined set of sentences that represented each sentiment. This set can be observed in Table X.

Sentiment	Sentence 1	Sentence 2
Anger	I am angry.	That is so annoying!
Disgust	I am disgusted.	That is repulsive!
Fear	I am frightened.	That is scary!
Joy	I am happy.	That is delightful!
Neutral	I am ok.	That is ok.
Sadness	I am sad.	That is so upsetting.
Surprise	I am surprised.	That is so amazing!

TABLE X
SENTENCES USED TO REPRESENT EACH SENTIMENT.

In order to evaluate the improvements achieved by our model (*SM*), we consider two baseline models: the baseline (*BL*) (DialoGPT-small); and the tag model (*Tag*), as a sentiment conditioned baseline (DialoGPT-small + sentiment tag).

				F1			
		PPL	**SES**	**m**	**m-NMC**	**M**	**M-NMC**
Dev	BL	92.4	16.8	42.5	15.6	13.2	6.4
	Tag	90.1	17.7	45.6	16.8	14.2	7.2
	SM	**86.3**	**18.0**	**62.7**	**42.3**	**29.1**	**22.4**
Test	BL	85.0	17.3	47.6	22.6	17.1	10.3
	Tag	**78.9**	16.7	61.4	43.9	24.2	17.4
	SM	79.4	**18.0**	**64.3**	**45.5**	**33.0**	**26.6**

TABLE XI
RESULTS OBTAINED ON THE DEVELOPMENT AND TEST SETS OF THE EMOTIONPUSH CORPUS WITH THREE DIFFERENT MODELS: DIALOGPT-SMALL (BL); DIALOGPT-SMALL + SENTIMENT TAG (TAG); AND DIALOGPT-SMALL + PRE-DEFINED SET OF SENTENCES (SM).

The results obtained by the three models, for both development and test sets, can be observed in Table XI. The *SM* model performs exceptionally well on the development set when compared to both the *BL* and the *Tag* approaches. Regarding the performance on the test set, despite the improvements not being as expressive, in particular, we achieve a better perplexity score on the *Tag* model, nonetheless, the *SM* is the model that performs better on this set as well, showing clear improvements on the sentiment metrics.

Following what was done for the EmotionPush corpus, we will analyse the results obtained on the *BL*, *Tag*, and *SM* models applied to the development and test sets of the DailyDialog corpus. The results can be observed in Table XII. Contrarily to the results obtained for the EmotionPush corpus, the *Tag* and the *SM* models perform more similarly on this dataset. On the development set, it is relevant to mention that the *Tag* model outperforms the *SM* model in the macro-F1 score by 4.5 points, and in the macro-F1 without the majority class by 6.8 points. On the test set, the *SM* model outperforms the *Tag* model on the sentiment metrics, more significantly on the macro metrics, but scores worse on the generation metrics.

As previously mentioned, the EmotionPush corpus is retrieved in an online chat context, which means the text is very informal, while the DailyDialog corpus was built from websites that are used to practice English, which makes the corpus more formal and fluent. This aspect could influence the quality of the generation models. In particular, the fact that the *Tag* and *SM* models fine-tuned with the DailyDialog corpus perform similarly could be an indication that the quality of the data used makes the models fine-tuned for this corpus not as dependent on the provided set of sentences, and a more simple option, such as a sentiment tag, is enough to guide the models. In contrast, the lower text quality of the EmotionPush corpus could be making the models fine-tuned for this dataset more reliant on full sentiment sentences in order to generate text conditioned on a sentiment.

				F1			
		PPL	**SES**	**m**	**m-NMC**	**M**	**M-NMC**
Dev	BL	9.7	28.5	82.7	25.7	20.9	9.4
	Tag	**9.3**	29.2	88.4	50.5	**42.5**	**34.0**
	SM	**9.3**	**29.5**	**88.5**	**50.6**	38.0	28.8
Test	BL	9.9	26.7	77.9	30.6	24.9	14.5
	Tag	**9.5**	**28.5**	83.6	51.2	42.6	34.6
	SM	9.6	27.3	**84.6**	**53.1**	**48.5**	**41.4**

TABLE XII
RESULTS OBTAINED ON THE DEVELOPMENT AND TEST SETS OF THE DAILYDIALOG CORPUS WITH THREE DIFFERENT MODELS: DIALOGPT-SMALL (BL); DIALOGPT-SMALL + SENTIMENT TAG (TAG); AND DIALOGPT-SMALL + PRE-DEFINED SET OF SENTENCES (SM).

D. Sentiment-Conditioned Conversational Agent

As mentioned in Section IV-B, given that using the sentiment of context sentences did not improve the reply sentiment prediction model's performance, the sentiment classification model is not part of the conversational agent, and it is only used as a metric to evaluate whether the generated sentences express the desired sentiment. Thus, the full setup includes: the reply sentiment prediction model, that receives the previous context of a conversation and outputs the appropriate sentiment for the conversational agent to express; and the text generation model, that given the predicted sentiment and the dialogue context, outputs a suitable reply.

In order to evaluate the sentiment-aware conversational agent, we will consider three systems: **BL**, the DialoGPT-small model, which is not conditioned on sentiment; **SM**, the DialoGPT-small + pre-defined set of sentences model. Since this model is conditioned on the gold sentiment label, it represents the proposed sentiment-aware conversational agent if the reply sentiment prediction model was perfect; and **FS**, the proposed sentiment-aware conversation agent.

				F1			
		PPL	**SES**	**m**	**m-NMC**	**M**	**M-NMC**
Dev	BL	92.4	16.8	42.5	15.6	13.2	6.4
	SM	**86.3**	**18.0**	**62.7**	**42.3**	**29.1**	**22.4**
	FS	88.1	16.0	49.2	21.1	15.6	8.2
Test	BL	85.0	17.3	47.6	22.6	17.1	10.3
	SM	**79.4**	**18.0**	**64.3**	**45.5**	**33.0**	**26.6**
	FS	80.3	16.6	49.8	21.6	15.7	8.4

TABLE XIII
RESULTS OBTAINED ON THE DEVELOPMENT AND TEST SETS OF THE EMOTIONPUSH CORPUS.

The results obtained for the development and test sets of the EmotionPush and DailyDialog corpora, can be observed in Tables XIII and XIV, respectively. For both datasets, the introduction of the reply sentiment prediction on the full

		PPL	SES	F1 m	F1 m-NMC	F1 M	F1 M-NMC
Dev	BL	9.7	28.5	82.7	25.7	20.9	9.4
	SM	**9.3**	**29.5**	**88.5**	**50.6**	**38.0**	**28.8**
	FS	9.5	28.4	83.7	25.6	20.4	8.7
Test	BL	9.9	26.7	77.9	30.6	24.9	14.5
	SM	**9.6**	27.3	**84.6**	**53.1**	**48.5**	**41.4**
	FS	9.7	**27.6**	77.9	30.8	26.0	15.8

TABLE XIV
RESULTS OBTAINED ON THE DEVELOPMENT AND TEST SETS OF THE DAILYDIALOG CORPUS.

system seems to be the bottleneck, given the lower performance achieved, namely on the sentiment metrics. This tells us that the sentiment being predicted by the reply sentiment prediction model is steering the model towards the wrong emotion, as expected by the results obtained for the reply sentiment prediction model. This proves the need for a better reply sentiment prediction model, which is key for the proper functioning of the proposed conversational agent. However, using the model with the pre-defined set of sentiment sentences in the full system still improves the *perplexity* when compared to the baseline, which shows that it is the set of sentences that is being concatenated to the input that is improving the *perplexity*, and not whether the model is receiving the correct sentiment or not. Regarding the sentence embedding similarity, on the EmotionPush corpus it is noticeable that the *SES* is better when the sentiment metrics and the *perplexity* achieve better results. On the DailyDialog corpus this is not as perceptive, given the low fluctuation of the metric between the evaluated models.

An example of the sentences generated by the developed sentiment-conditioned text generation model (*SM*) can be observed in Table XV. For the same context sentence we present the sentences generated by the model conditioned on each of the target sentiments. Additionally, we also show the sentence generated by the baseline, which is not conditioned on any sentiment. We can observe that the model is capable of generating fluent and adequate answers for all sentiments. Furthermore, most sentences generated express the desired sentiment. Another interesting aspect about the generated sentences is that the model only repeated keywords from the pre-defined set of sentiment sentences twice on a sentence generated conditioned on the *fear* emotion.

Example 1	
Context	How do you feel?
Baseline	I feel great, thanks.
Anger	I'm sick of this city.
Disgust	I'm very nauseous. I've never felt so nauseous.
Fear	I'm scared. I'm scared of the future.
Joy	I like it very much.
Sadness	I'm really sorry, I'm really not feeling well.
Surprise	Oh, my god! My eyes are burning!

TABLE XV
EXAMPLE OF SENTENCES GENERATED BY THE SENTIMENT-CONDITIONED TEXT GENERATION MODEL USING THE PRE-DEFINED SET OF SENTENCES GIVEN THE SAME CONTEXT. THE WORDS IN RED REPRESENT WORDS THAT ARE PRESENT IN THE PRE-DEFINED SET OF SENTENCES.

An example of a generated dialogue between the proposed sentiment-aware conversational agent and a user can be observed in Table XVI. The text generation model seems to be able to generate the sentiments it is conditioned to, and also to consider the past context, with the only mistake being on the second chatbot sentence, where the model confused who called the police. Again, in this dialogue the sentence generated with the *fear* emotion repeated a keyword from the pre-defined set of sentiment sentences. Interestingly, the *fear* sentiment is the most underrepresented sentiment in the corpus, which might have made the model rely more on the provided set.

Dialogue Example		
User:		I just called the police, I think someone broke into our home while we were at work.
Chatbot:	[FEAR]	I am really scared.
User:		Me too! What do we do now?
Chatbot:	[JOY]	I'm really glad that I called the police. I really wanted to go home.
User:		I just want this nightmare to be over.
Chatbot:	[ANGER]	Oh, it's not over. We are going to have to go home now.

TABLE XVI
EXAMPLE OF A DIALOGUE BETWEEN A USER AND THE PROPOSED SENTIMENT-AWARE CONVERSATIONAL AGENT. [SENTIMENT] REPRESENTS THE SENTIMENT PREDICTED BY THE REPLY SENTIMENT PREDICTION MODEL AND THAT IS USED TO CONDITION THE REPLY OF THE CONVERSATIONAL AGENT ON THAT UTTERANCE. THE WORDS IN RED REPRESENT WORDS THAT ARE PRESENT IN THE SET OF SENTENCES.

E. Human Evaluation

As mentioned in [14], automatically evaluating empathetic conversational agents is a challenging task given the limitations of automatic metrics. In particular, the most common text generation metrics evaluate the word/lexical overlap between the gold and generated sentences, and in a dialogue setting there can be many correct answers. Furthermore, the experiments previously reported further motivate the challenge of relying on these metrics to evaluate sentiment-aware conversational agents: giving to the model the exact same context and condition it on different sentiments drastically changes the outcome of the generated sentences. For that matter, human evaluation became a popular method to evaluate conversational agents. For this evaluation we were able to gather answers from seven annotators with a proficient English level.

The most important aspects to evaluate regarding the generated text were the adequacy given the previous context, and whether it expressed the desired sentiment. This evaluation was done by sampling at random 40 inputs from the test set of each corpus and retrieving the corresponding replies generated by four of the developed architectures: **BL**, the DialoGPT-small model; **SB**, the DialoGPT-small + *tag* setup. This model allows us to have a baseline that is conditioned on a sentiment; **SM**, the DialogGPT-small + pre-defined set of sentences; and **FS**, the proposed sentiment-aware conversational agent, with the reply sentiment prediction and text generation models.

One detail about the chosen architectures is that both the SB and the SM approaches use the gold sentiment label to condition the sentiment of the generated sentence. In that sense, these approaches can be considered as being in the sentiment-aware conversational agent scenario, where the reply sentiment prediction model works perfectly.

In order to evaluate the adequacy of the reply we asked the annotators the following question: "Do the replies sound

appropriate considering the context of the dialogue?". A similar process was followed to evaluate the sentiment of the sentences. Our goal with this evaluation was to assess if the model was able to generate sentences with a desired sentiment. The question asked to the annotators was "Do the replies represent the `<sentiment_name>` emotion?". In this evaluation we asked the annotators specifically to not consider the previous context of the replies. Additionally, given the multitude of sentiments present that often can be interchanged, we also asked them to consider whether the sentence being evaluated could be said to express the asked sentiment. For example, "What?" could be used to express *anger* or *surprise*, depending on the tone used. We use a 2-point Likert scale for both questions.

	EmotionPush		*DailyDialog*	
Model	**Adequacy**	**Sentiments**	**Adequacy**	**Sentiments**
BL	0.4292	0.325	0.4958	0.3708
SB	0.5042	0.5167	**0.6542**	0.7084
SM	**0.6167**	**0.6583**	0.6292	**0.7625**
FS	0.3917	0.3542	0.5667	0.3292

TABLE XVII
AVERAGE EMOTIONPUSH AND DAILYDIALOG SCORES.

The results obtained for the human evaluation performed on the dialogues sampled from the EmotionPush corpus can be observed in Table XVII. Given that we are using a Likert scale of 2-points the reported scores correspond to the ratio of positive answers to a given question. E.g., an adequacy score of 0.6 for the *BL* model means that 60% of the generated sentences by this model were considered adequate. The *sentiments* score corresponds to the ratio of positive answers considering all sentiments. It is clear that the model that achieves the best performance was the *SM* model. In particular, we highlight how this model improved the adequacy of the replies when compared to all other models. It also achieves the highest sentiments scores. It is also interesting to observe that, despite the accumulated error of the *FS* due to the reply sentiment prediction model, the *FS* still outperforms the *BL* by 0.0292 points on the sentiments metric. Nonetheless, it achieves a worse performance on the adequacy metric. There also seems to exist a correlation between the sentiments metric and how adequate the replies are. The models that achieved a higher sentiment metric, tend to also be more adequate.

Regarding the DailyDialog corpus, both *SB* and *SM* approaches perform similarly. The improvements of both models when compared to the *BL* are considerable across all metrics.

We finalize this analysis with the correlation between the obtained automatic metrics, and the human evaluation by using the Pearson correlation to measure the linear relationship between them. In particular, we correlate four points, corresponding to pairs of the automatic and human metrics obtained for each evaluated model. It is important to mention that we should take into consideration that a correlation using four points is not ideal, and might not lead to statistically significant values. We can observe the aforementioned correlation for the EmotionPush and DailyDialog corpora in Table XVIII. On the EmotionPush corpus, the correlation between the human evaluation metrics and the perplexity is very low. Regarding the same correlation on the DailyDialog corpus, we can observe that the correlation between the perplexity and the sentiments metric is high, while between the perplexity and the adequacy is close to perfect. This highlights the difference in the quality of the text in both datasets. Regarding the correlation between the human evaluation metrics and the sentence embedding similarity, on the EmotionPush corpus is close to perfect on both metrics, while on the DailyDialog the correlation between the SES and the Adequacy is high, but between the SES and the sentiments metric is lower. This could be related to the fact on this evaluation the *SM* performs better on the sentiments metric, while on the automatic evaluation, we saw that the *SB* model performed better on the SES metric. Finally, we can observe that the correlation between the sentiment automatic metrics and the human evaluation metrics is also high for both corpora. It is relevant to highlight that models fine-tuned with formal and fluent data, such as the DailyDialog corpus, seem to perform well with simpler sentiment-conditioning approaches. In contrast, models fine-tuned with informal data, such as the EmotionPush corpus, seem to rely more on full sentiment sentences in order to perform well. This hypothesis is further validated with these experiments, since the annotators gave higher scores to the *SM* model fine-tuned for the EmotionPush corpus, while the annotations gathered for the DailyDialog corpus showed similar results for the *SM* and *SB* approaches.

	PPL	**SES**	**m-NMC**	**M-NMC**
		EmotionPush		
Adequacy	0.18	0.9524	0.8015	0.8522
Sentiments	0.01	0.9519	0.7992	0.8328
		DailyDialog		
Adequacy	-0.981	0.7992	0.8852	0.8589
Sentiments	-0.7345	0.4475	0.9966	0.9839

TABLE XVIII
PEARSON CORRELATION BETWEEN THE AUTOMATIC METRICS (PERPLEXITY, SENTENCE EMBEDDING SIMILARITY, MICRO/MACRO-F1 WITHOUT THE MAJORITY CLASS) AND THE HUMAN EVALUATION METRICS (ADEQUACY AND SENTIMENTS) APPLIED TO THE EMOTIONPUSH AND DAILYDIALOG DATASETS.

V. CONCLUSION AND FUTURE WORK

In this work we explored reply sentiment prediction and conditioned text generation as a way to build a sentiment-aware conversational agent. In particular, we saw that using multiple context sentences on the input of the reply sentiment prediction model, and using a pre-defined set of sentiment sentences to condition the text generation model, improved performance when compared to baseline models. Furthermore, for text generation, we showed how our approach resulted in clear gains for small datasets. Additionally, we observed how the reply sentiment prediction model is the bottleneck of the full system. Finally, we also performed a human evaluation on the developed models which corroborated the results obtained on the automatic evaluation.

As future work, we consider that improving the reply sentiment prediction model is crucial for a better performance of the conversational agent. To do so, we propose to explore different ways to incorporate the retrieval augmentation methods [55], prompt-based learning [56], or hybrid approaches with both data-driven and pre-defined rules.

REFERENCES

[1] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[2] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[3] O. Vinyals and Q. Le, "A neural conversational model," 2015.

[4] L. Shang, Z. Lu, and H. Li, "Neural responding machine for short-text conversation," in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 2015, pp. 1577–1586.

[5] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," 2017.

[6] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," 2018.

[7] H. Rashkin, E. M. Smith, M. Li, and Y.-L. Boureau, "Towards empathetic open-domain conversation models: A new benchmark and dataset," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019, pp. 5370–5381.

[8] N. Miao, Y. Song, H. Zhou, and L. Li, "Do you have the right scissors? tailoring pre-trained language models via monte-carlo methods," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 3436–3441.

[9] H. Zhou, M. Huang, T. Zhang, X. Zhu, and B. Liu, "Emotional chatting machine: Emotional conversation generation with internal and external memory," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[10] C. Huang, O. R. Zaiane, A. Trabelsi, and N. Dziri, "Automatic dialogue generation with expressed emotions," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, 2018, pp. 49–54.

[11] T. Wolf, V. Sanh, J. Chaumond, and C. Delangue, "Transfertransfo: A transfer learning approach for neural network based conversational agents," *arXiv preprint arXiv:1901.08149*, 2019.

[12] S. Dathathri, A. Madotto, J. Lan, J. Hung, E. Frank, P. Molino, J. Yosinski, and R. Liu, "Plug and play language models: a simple approach to controlled text generation," *arXiv preprint arXiv:1912.02164*, 2019.

[13] N. S. Keskar, B. McCann, L. R. Varshney, C. Xiong, and R. Socher, "Ctrl: A conditional transformer language model for controllable generation," *arXiv preprint arXiv:1909.05858*, 2019.

[14] Y. Xie and P. Pu, "Generating empathetic responses with a large scale dialog dataset," *arXiv preprint arXiv:2105.06829*, 2021.

[15] M. Davidow, "Organizational responses to customer complaints: What works and what doesn't," *Journal of service research*, vol. 5, no. 3, pp. 225–250, 2003.

[16] T. Hu, A. Xu, Z. Liu, Q. You, Y. Guo, V. Sinha, J. Luo, and R. Akkiraju, "Touch your heart: A tone-aware chatbot for customer care on social media," in *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, 2018, pp. 1–12.

[17] X. Li and M. Zhang, "Emotion analysis for the upcoming response in open-domain human-computer conversation," in *Asia-Pacific Web (APWeb) and Web-Age Information Management (WAIM) Joint International Conference on Web and Big Data*. Springer, 2018, pp. 352–367.

[18] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," 2019.

[19] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized bert pretraining approach," 2019.

[20] C. Sun, X. Qiu, Y. Xu, and X. Huang, "How to fine-tune bert for text classification?" 2020.

[21] J. Kim, H. Ko, S. Song, S. Jang, and J. Hong, "Contextual augmentation of pretrained language models for emotion recognition in conversations," in *Proceedings of the Third Workshop on Computational Modeling of People's Opinions, Personality, and Emotion's in Social Media*, 2020, pp. 64–73.

[22] Y.-H. Huang, S.-R. Lee, M.-Y. Ma, Y.-H. Chen, Y.-W. Yu, and Y.-S. Chen, "Emotionx-idea: Emotion bert–an affectional model for conversation," *arXiv preprint arXiv:1908.06264*, 2019.

[23] W. Shen, S. Wu, Y. Yang, and X. Quan, "Directed acyclic graph network for conversational emotion recognition," *arXiv preprint arXiv:2105.12907*, 2021.

[24] J. Herzig, G. Feigenblat, M. Shmueli-Scheuer, D. Konopnicki, A. Rafaeli, D. Altman, and D. Spivak, "Classifying emotions in customer support dialogues in social media," in *Proceedings of the 17th annual meeting of the special interest group on discourse and dialogue*, 2016, pp. 64–73.

[25] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.

[26] C. Bothe, S. Magg, C. Weber, and S. Wermter, "Dialogue-based neural learning to estimate the sentiment of a next upcoming utterance," in *International Conference on Artificial Neural Networks*. Springer, 2017, pp. 477–485.

[27] Z. Wen, J. Cao, R. Yang, S. Liu, and J. Shen, "Automatically select emotion for response via personality-affected emotion transition," in *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, 2021, pp. 5010–5020.

[28] A. Mehrabian, "Pleasure-arousal-dominance: A general framework for describing and measuring individual differences in temperament," *Current Psychology*, vol. 14, no. 4, pp. 261–292, 1996.

[29] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," *Advances in neural information processing systems*, vol. 27, pp. 3104–3112, 2014.

[30] X. Zhou and W. Y. Wang, "Mojitalk: Generating emotional responses at scale," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2018, pp. 1128–1137.

[31] P. Colombo, W. Witon, A. Modi, J. Kennedy, and M. Kapadia, "Affect-driven dialog generation," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 2019, pp. 3734–3743.

[32] S. Santhanam and S. Shaikh, "Emotional neural language generation grounded in situational contexts," in *Proceedings of the 4th Workshop on Computational Creativity in Language Generation*, 2019, pp. 22–27.

[33] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners," *OpenAI blog*, vol. 1, no. 8, p. 9, 2019.

[34] N. Asghar, P. Poupart, J. Hoey, X. Jiang, and L. Mou, "Affective neural response generation," in *European Conference on Information Retrieval*. Springer, 2018, pp. 154–166.

[35] A. Yadollahi, A. G. Shahraki, and O. R. Zaiane, "Current state of text sentiment analysis from opinion to emotion mining," *ACM Computing Surveys (CSUR)*, vol. 50, no. 2, pp. 1–33, 2017.

[36] J. Li and X. Sun, "A syntactically constrained bidirectional-asynchronous approach for emotional conversation generation," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018, pp. 678–683.

[37] L. Xu, H. Lin, Y. Pan, H. Ren, and J. Chen, "Constructing the affective lexicon ontology," *Journal of the China society for scientific and technical information*, vol. 27, no. 2, pp. 180–185, 2008.

[38] P. Zhong, D. Wang, and C. Miao, "An affect-rich neural conversational model with biased attention and weighted cross-entropy loss," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, 2019, pp. 7492–7500.

[39] N. Lubis, S. Sakti, K. Yoshino, and S. Nakamura, "Eliciting positive emotion through affect-sensitive dialogue response generation: A neural network approach," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018.

[40] H. Rashkin, E. M. Smith, M. Li, and Y.-L. Boureau, "I know the feeling: Learning to converse with empathy," 2018.

[41] E. Dinan, S. Roller, K. Shuster, A. Fan, M. Auli, and J. Weston, "Wizard of wikipedia: Knowledge-powered conversational agents," *arXiv preprint arXiv:1811.01241*, 2018.

[42] N. Moghe, S. Arora, S. Banerjee, and M. M. Khapra, "Towards exploiting background knowledge for building conversation systems," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018.

[43] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel *et al.*, "Retrieval-augmented generation for knowledge-intensive nlp tasks," *arXiv preprint arXiv:2005.11401*, 2020.

[44] R. P. Ramos, P. Pereira, H. Moniz, J. P. Carvalho, and B. Martins, "Retrieval augmentation for deep neural networks," *arXiv preprint arXiv:2102.13030*, 2021.

[45] N. Reimers and I. Gurevych, "Sentence-bert: Sentence embeddings using siamese bert-networks," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 11 2019. [Online]. Available: https://arxiv.org/abs/1908.10084

[46] J. Johnson, M. Douze, and H. Jégou, "Billion-scale similarity search with gpus," *IEEE Transactions on Big Data*, 2019.

[47] C.-C. Hsu, S.-Y. Chen, C.-C. Kuo, T.-H. Huang, and L.-W. Ku, "EmotionLines: An emotion corpus of multi-party conversations," in *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*. Miyazaki, Japan: European Language Resources Association (ELRA), May 2018. [Online]. Available: https://www.aclweb.org/anthology/L18-1252

[48] Y. Li, H. Su, X. Shen, W. Li, Z. Cao, and S. Niu, "Dailydialog: A manually labelled multi-turn dialogue dataset," 2017.

[49] P. Ekman, "An argument for basic emotions," *Cognition & emotion*, vol. 6, no. 3-4, pp. 169–200, 1992.

[50] W. Falcon and The PyTorch Lightning team, "PyTorch Lightning," 3 2019. [Online]. Available: https://github.com/PyTorchLightning/pytorch-lightning

[51] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, C. Ma, Y. Jernite, J. Plu, C. Xu, T. Le Scao, S. Gugger, M. Drame, Q. Lhoest, and A. M. Rush, "Transformers: State-of-the-Art Natural Language Processing." Association for Computational Linguistics, 10 2020, pp. 38–45. [Online]. Available: https://www.aclweb.org/anthology/2020.emnlp-demos.6

[52] J. Howard and S. Ruder, "Universal language model fine-tuning for text classification," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2018, pp. 328–339.

[53] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[54] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[55] X. Wang, Y. Jiang, N. Bach, T. Wang, Z. Huang, F. Huang, and K. Tu, "Improving named entity recognition by external context retrieving and cooperative learning," *arXiv preprint arXiv:2105.03654*, 2021.

[56] X. Han, W. Zhao, N. Ding, Z. Liu, and M. Sun, "Ptr: Prompt tuning with rules for text classification," *arXiv preprint arXiv:2105.11259*, 2021.