Creating an Emotion Responsive Dialogue System

by

Ankit Vadehra

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Mathematics
in
Computer Science

Waterloo, Ontario, Canada, 2018

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

The popularity of deep neural networks and vast amounts of readily available multi-domain textual data has seen the advent of various domain/task specific and domain agnostic dialogue systems. In our work we present a general dialogue system that can provide a custom response based on the emotion or sentiment label selected. A dialogue system that can vary its response based on different affect labels can be very helpful for designing help-desk or social help assistant systems where the response has to follow a certain affective tone, such as positive, compassionate, etc.

To address this task, we design a model that can generate coherent response utterances conditioned on a specified affect label (emotion or sentiment). We use a Sequence-to-Sequence model with an adversarial objective to remove affect from the learned representation of the input utterance, and generate the response based on this representation and the target affect label. Two models were evaluated: affect embedding and multi-decoder. We hypothesize that removal of the affect from the input utterance is helpful in generating a response conditioned on a different affect label. The models were evaluated on a large Twitter dialogue corpus. The results support our hypothesis.

Acknowledgements

I would primarily like to thank my supervisor Prof. Olga Vechtomova who was always very supportive and helpful throughout my Masters degree and provided me with numerous invaluable suggestions regarding my research work and finding an effective and interesting thesis topic. I'd especially like to thank Prof. Olga for introducing me to the incredible research that utilizes Deep Neural Network models for solving various Natural Language Processing tasks.

I would also like to thank Prof. Pascal Poupart and Prof. Gordon Cormack for agreeing to read my thesis and for the work I did with and under their guidance which was really fun and informative.

There are multiple people, both peers and faculty members, at the University of Waterloo who are inadvertently responsible for my successful completion of the Masters program in Computer Science. I'd like to thank them for all the random/focused discussions, the high-fives, the fist-bumps and the colloquial greetings. I'd especially like to thank Hareesh Bahuleyan and Vineet John for their insightful suggestions about my project.

I'd also like to thank three Engineering Undergraduate Research Assistants, Anant Kandadai, Junn Hei Jonathan Cho and Michael Wang, for annotating the TV scripts datasets.

I'd be remiss if I didn't mention my appreciation for the CS-Grad administrative staff especially Marie Kahkejian and Greg Mctavish who were always helpful in answering and solving all of my queries. Also, setting up and managing machines can be problematic sometimes, so, I'd like to thank Gordon Boerke for always helping me out with my system.

Finally, I'd like to thank my family *(my very own North Star)*, especially my parents *(who are amazingly awesome!)* for everything, because *familia supra omnia*.

Dedication

To the *insane* amount of amazing digital media content,
that kept me *sane*.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Dialogue systems, conversational agents and chatbots are a well researched area in NLP. There has been a plethora of research trying to create domain/task specific and domain agnostic chatbots. Customer support or food ordering chatbots are examples of task specific conversational agents where the success of the system is determined by the ability of the agent to complete its task [70]. Siri, Cortana and Google Assistant are domain agnostic agents that are able to converse about multi-domain general topics. There are some dialogue systems that can actually condition their response on specific attributes like sentiment, emotion, different personalities, etc. [95, 45, 52].

1.1 Problem Definition

Dialogue systems can generate text utterance in response to an input utterance. An emotion conditioned dialogue system can generate diverse and varied response to the same utterance based on different emotions selected. A system that can generate specific emotion based responses can be hugely beneficial for creating dialogue systems that are more interactive and able to perceive the utterance tone and maintain longer conversations [49].

1.2 Challenges

1.2.1 Dialogue System

Most of the techniques for dialogue systems utilize the research in Machine Translation and Question Answering systems [79]. But the difference in the two domains is that a translation system usually has a one-to-one mapping between each word in the source and target sentence, whereas, a dialogue system can have multiple possible responses to the same utterance which often makes it generate vague and short responses. [86]

1.2.2 Emotion Domain

Even though we use the term affect to encompass both sentiment and emotion; detailed research states that there is a difference between affect, feelings, emotions, sentiments, and opinions from a psychological perspective. Concentrating on the sentiment and emotion domain we find that emotion is very hard to predict from text alone and sentiment is different from emotion as it involves forming opinions over a longer period of time [55]. There are various emotion models which we describe in detail in 2.3. For our experiments we utilize the Ekman emotion model [22].

Even though our primary goal is to design an emotion responsive dialogue system, we carry out experiments on both the sentiment and emotion spectrum. Sentiment is a two-class (positive and negative) problem and generally has clear class separating markers and hence a good indicator to check whether the approach is working or not. On the other hand, emotion is a six-class (anger, disgust, fear, joy, sadness and surprise) problem and harder to differentiate between.

To verify the hypothesis that emotion is difficult to differentiate, we took the four most popular cross-domain emotion annotated corpora, generated a $1:3(test:train)$ split and used Naive Bayes and SVM classifiers [1] with bag-of-words and TF-IDF n-gram features [82] to see how well emotion transfers between different domains. We also performed human annotation agreement evaluation using Cohen's kappa score [13] on dialogue transcripts for the TV Series F.R.I.E.N.D.S[1] and LOST[2]. F.R.I.E.N.D.S is primarily a comedy series and we did not find many utterances for anger, disgust and fear in the random selection of utterances to annotate. Since LOST is a drama/suspense series we assumed that it would

[1] http://www.livesinabox.com/friends/scripts.shtml
[2] http://lostpedia.wikia.com/wiki/Portal:Transcripts

be a better estimate to see human annotation agreement for the six primary emotions, i.e. anger, disgust, fear, joy, sadness and surprise.

Emotion Dataset	Emotion Label Count							
	anger	disgust	fear	guilt	joy	sadness	shame	surprise
ISEAR	1095	1095	1094	1092	1093	1095	1095	-
FairyTales*	219	-	167	-	446	265	-	115
NRCTEC	1556	762	2817	-	8241	3831	-	3850
BlogPosts*	180	173	116	-	537	174	-	116

Table 1.1: Emotion Datasets Evaluated

Experiment	Test Set	Train Set
Exp. 1	ISEAR	FairyTales, NRCTEC, BlogPosts
Exp. 2	BlogPosts	FairyTales, NRCTEC, ISEAR
Exp. 3	FairyTales	ISEAR, NRCTEC, BlogPosts

Table 1.2: Experiments run on the emotion datasets.

The emotion datasets used to check the cross-domain transferability and the experiments run are described in Table 1.1 (*subset of the dataset with high annotator agreement) and Table 1.2. We explain the datasets used - ISEAR [68], FairyTales [2], NRCTEC [53], BlogPosts [3] in Section 2.3.2.

Evaluation Results

Multiple runs were performed by selecting different feature/model/extraction method.

Annotator	Score
a $\leftrightarrow$ b	0.303
c $\leftrightarrow$ d	0.110
d $\leftrightarrow$ e	0.1977
c $\leftrightarrow$ e	0.3124
Average	0.231

Table 1.3: Human annotation evaluation between 5 annotators on LOST transcript using Cohen's Kappa score.

Test Dataset	Model, Feature & N-Gram Group	Accuracy
ISEAR	SVM-FC(1,1)	0.431
BlogPosts	MNB-FC(1,1)	0.535
FairyTales	SVM-TFIDF(1,1)	0.536

Table 1.4: Highest Accuracy Groups for Different Features

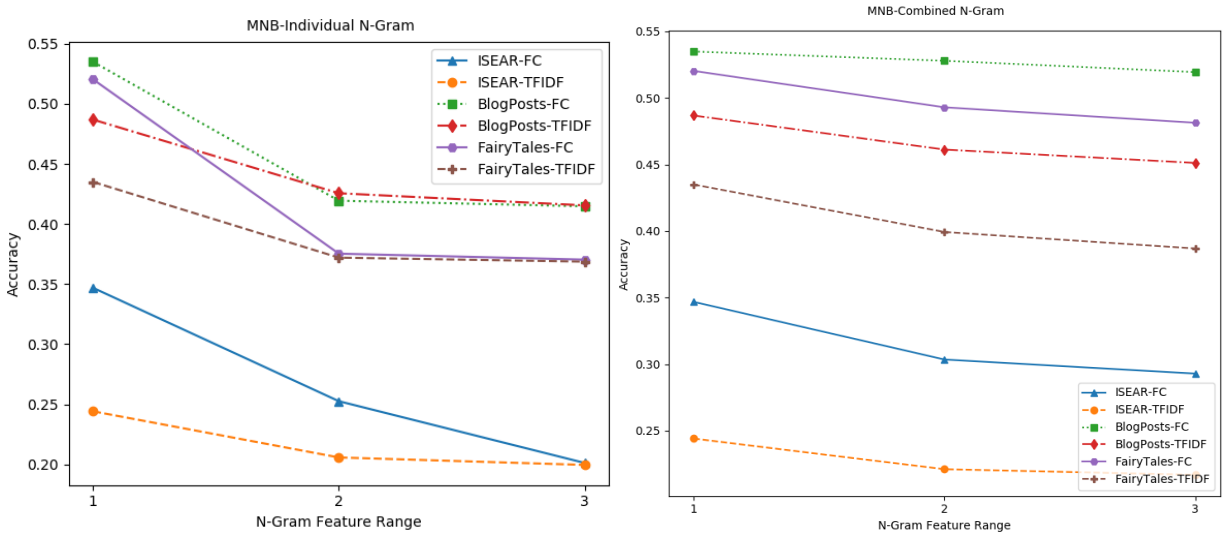
Results Notations:
• **Individual N-Gram Group** represents N-Gram group taken individually. For example 1: unigram, 2: bigram, 3: trigram and so on.
• **Combined N-Gram Group** represents taking all combinations from 1 to that group. N-Gram Group $i = \Sigma_{n=1}^{i}$ n-gram. For example 1: Unigram, 2: Unigram+Bigram, 3:Unigram+Bigram+ Trigram and so on.
• **N-Gram Group:** (i, j) = $\Sigma_{n=i}^{j}$ n-gram
For example (1,1) : unigram, (2,2) : Bigram, (1,2) : unigram + bigram, (3,3) : trigram, (1,3) : unigram+bigram+trigram.. etc.
• **Learner Models:** SVM : Support Vector Machine, MNB : Multinomial Naive Bayes.
• **Feature:** FC : Bag-of-Words, TFIDF : Term Frequency-Inverse Document Frequency.

The detailed classifier accuracy results are plotted in Figure 1.1. Figure 1.1a-1.1b show the results for the three experiments using the MNB classifier and Figure 1.1c-1.1d shows the results using an SVM classifier with bag-of-words and TF-IDF features for both classifiers. The highest results for each test experiment are tabulated in Table 1.4.

We see a very poor annotator mutual agreement average of 0.231 (Table 1.3) and the classifiers show poor emotion classification result. Also, we noticed in our experiments that excluding the Twitter-NRCTEC emotion dataset from the training set results in poor performance. Hence, we plan on utilizing Twitter as the data source for our dialogue system. The results obtained agree with the research and our assumption that it is very hard to distinguish between emotions in text and that emotion does not transfer well across domains.

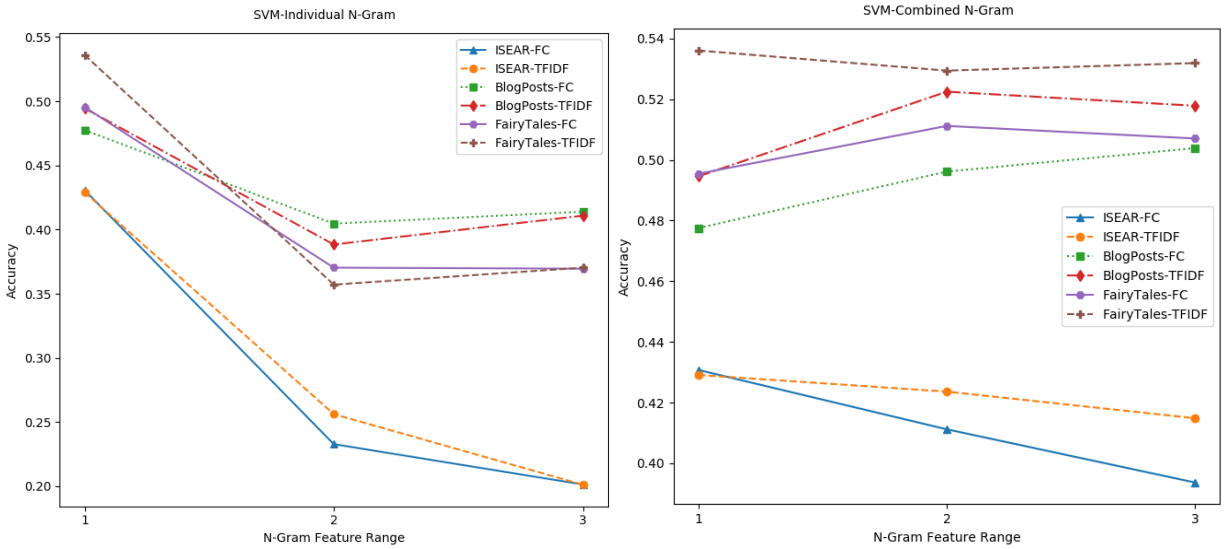
1.2.3 Dialogue Dataset

Since Twitter is an excellent source of gathering vast amount of informal, cross-domain dialogue data we use Twitter to gather our data. Unfortunately, the informal and improper grammar coupled with short text length dialogues makes it difficult to learn an accurate



(a) MNB-Individual N-Gram

(b) MNB-Combined N-Gram



(c) SVM-Individual N-Gram

(d) SVM-Combined N-Gram

Figure 1.1: Cross-domain emotion transferability results.

syntactical language model.

1.3 Approach

In a typical dialogue system model that utilizes the encoder-decoder framework, the encoder generates the latent representation of the source utterance and the decoder uses this representation to generate the response utterance. It is our hypothesis that during the dialogue system training if the first utterance is highly emotional it will inadvertently bias and direct the emotion of the response utterance. This makes generating varied response conditioned on different aspect attributes difficult.

We aim to utilize adversarial learning to separate the context from the affect in the first utterance to circumvent this problem. We try to make the latent representation learned by the encoder void of any affect characteristic while retaining the content information. This way the decoder is able to learn to generate varied emotional responses based on different affect criteria parameters.

We utilize an emotion classifier [17] trained on vast amount of Twitter data to classify our Twitter gathered dataset and then we separate the emotion into the appropriate sentiment and initially try to create a model that can generate response conditioned on sentiment (positive/negative). We then extend those models to work on the six-class emotion domain.

1.4 Contribution

We gather and present a multi-turn dialogue dataset that has been emotion tagged utilizing pre-trained emotion classifier model. We also propose a new method of introducing adversarial training in a dialogue system to better condition the response on the chosen affect attribute. This technique might have vast uses in designing other personalized and attribute conditioned dialogue systems.

1.5 Thesis Layout

We present detailed background information and literature survey on each of our models' individual components in Chapter 2. We provide a brief history and current research direction of conversation agents and dialogue systems in 2.2. The emotion and sentiment analysis problem is presented and explained in 2.3. We present the different emotion annotated datasets and the classifier models trained on those datasets in 2.4. We describe

and explain controlled and conditioned text generation in 2.5. We describe our methods, data collection and annotation process in Chapter 3. Chapter 4 describes the results obtained by the models trained on the sentiment and emotion datasets. We summarize our work and provide some future directions in Chapter 5.

Chapter 2

Background and Literature Survey

We use the first section of this chapter to explain some of the components and building blocks used in our model. We use the subsequent sections of this chapter to discuss the related work in this area.

2.1 Basic Model Components

2.1.1 Feed Forward Neural Network

We utilize fully connected feed forward neural networks in our work, which are unidirectional neural networks without cycles/recurrent loops between intermediate layers and neurons [69]. Single layer and multi layer perceptron models are two types of feed forward networks. Single layer perceptron network does not have any intermediate layers between the input and output layer, whereas, the multilayer perceptron can have one or more intermediate layers between the input and output layer. The model has weights assigned to the all the layers which are updated during training to allow for proper convergence. The layers can also have non-linear activation functions like tanh and ReLU to allow for training more complex mappings.

Feed forward neural networks are often used for transforming the dimension of inputs and outputs of different complex models like convolutional neural network (CNN) and recurrent neural network (RNN). A softmax function on the output of a feed forward neural network produces a probability distribution over the outputs with sum 1 that can

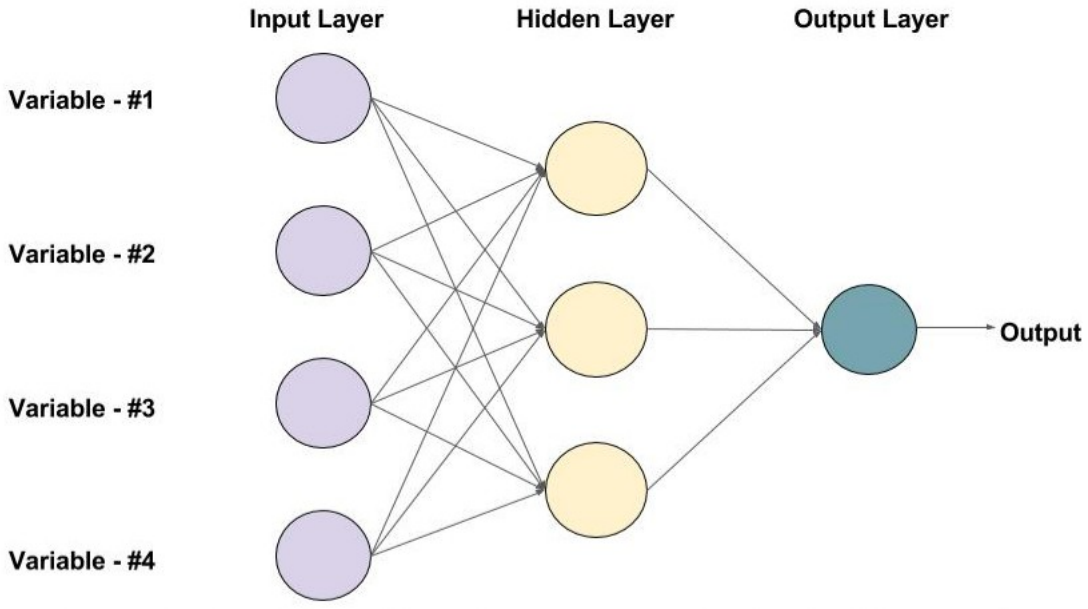


Figure 2.1: A multi-layer/feed-forward neural network model.

be interpreted as confidence scores for the classes and can be used for training a multi-class classifier. An example of a feed-forward neural network with a single hidden layer consisting of three neurons is given in Figure 2.1.

2.1.2 Word Embeddings

Word embeddings are $n - dimensional$ distributional representations of words in a continuous vector space calculated by utilizing the surrounding words [30, 16]; the idea being that a word can be categorized by the company it keeps [26]. If $W = \{w_1, w_2, ..., w_n\}$ is the set of all words in a dataset then $E(w_i) \in \mathbb{R}^{n \times D}$ is a high dimensional mapping for each w_i in a continuous $D - dimensional$ vector space [80] called the embedding space and n is the number of mappings.

There are various shallow neural network models that are used to compute the embedding space like GloVe [60], Word2Vec [51], fastText [8]. Vector representations of words are useful in improving various language modeling and syntactical tasks like classification, syntactic parsing, sentiment analysis [75, 76]. The representation of different words in a vector space is able to capture the relationship between words [51] and syntactically and semantically similar words are found closer to each other in the embedding space.

2.1.3 Recurrent Neural network (RNN)

RNN is a type of neural network models which can handle variable length data. Each RNN unit has a self loop and an internal state which is helpful in processing sequential data like audio waveform, stock value, text and video. RNNs are helpful in processing sequential data because they can utilize each individual data point as well as the relation of that data point with the preceding data points, which results in generating a more comprehensive and proper textual representation. As a result of their usefulness, RNNs are used in NLP for various tasks like text classification, language generation, named entity recognition, etc [91].

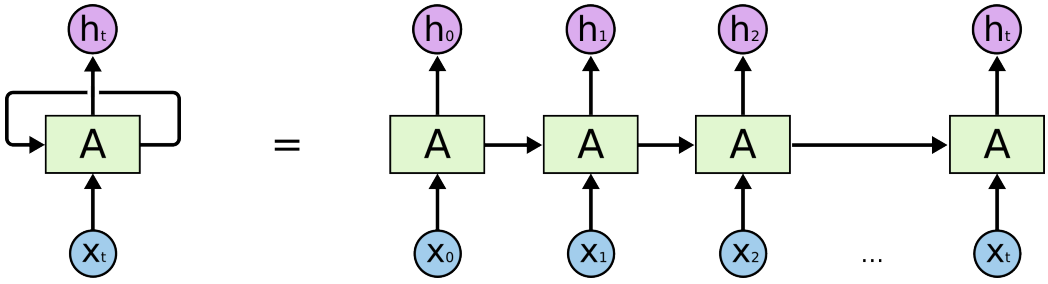


Figure 2.2: An un-rolled Recurrent Neural Network (RNN)[1].

Figure 2.2 shows an RNN network. x_i denotes the input to an RNN which is usually the embedding vector of a word from from the vocabulary. h_i denotes the hidden state carried forward and outputted at teach time-step by the RNN. This figure denotes the simplest form of recurrent network. In practice we often use a Long Short Term Memory (LSTM) [34] or Gated Recurrent Unit (GRU) [19] network. Both these network are complex and perform much better at language tasks. They consist of internal state and gates combined with non-linearities to allow much better learning of complex functions and mappings.

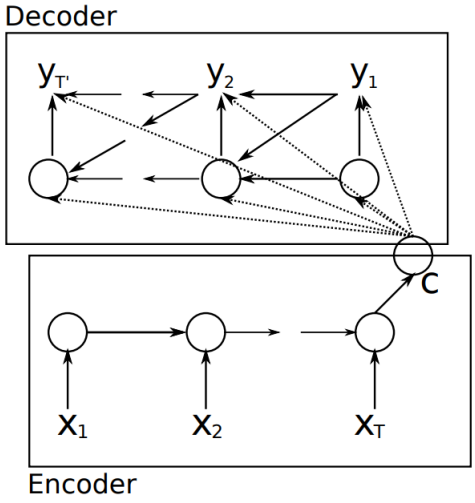
2.1.4 Sequence-to-Sequence (Encoder-Decoder) with Attention

A sequence-to-sequence (Seq2Seq) model utilizes the encoder-decoder architecture made up of RNNs to learn mappings that can generate an entire sequence of tokens. The architecture was designed and inspired to solve the machine translation problem [79]. If a dataset is comprised of source and target utterances then the encoder generates the latent

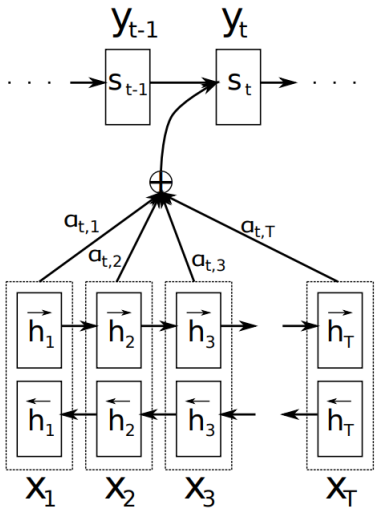
[1] http://colah.github.io/posts/2015-08-Understanding-LSTMs/

representation of the source utterance. The decoder utilizes the representation to generate the target sequence.

Figure 2.3a shows an example of a Seq2Seq architecture. $X = \{X_1, X_2, ..., X_T\}$ is a variable length input to the encoder network. The input is generally the embedding representation for each distinct word in the vocabulary. The encoder RNN takes the input embeddings and generates the latent representation of the whole sentence represented by C. The latent representation is given to the decoder RNN which tries to generate a variable length output $Y = \{Y_1, Y_2, ..., Y_T\}$. The latent representation C is utilized differently by different architectures. The Seq2Seq model introduced by Sutskever et al. uses the latent representation C to initialize the initial hidden state of the decoder RNN h_0 [79]. Other models concatenate the representation vector C to the model input at each timestep. Cho et al. (Figure 2.3a) present a model where C is passed to the RNN model at each timestep. In our model (Section 3.4) we use the model where the source utterance representation vector is passed to the decoder at each timestep.



(a) A Seq2Seq network by Cho et al. [12]



(b) A Seq2Seq network with attention by Bahdanau et al. [5]

Figure 2.3: A Sequence-to-Sequence (Seq2Seq) Network with and without attention.

There have been various modifications to improve performance of the traditional Seq2Seq model. During implementation it was observed that reversing the source sentence during

encoding as well as passing each source sentence twice during training significantly improved performance [92]. However, these techniques can be categorized as 'practical hacks' that improve performance while designing the model. Attention mechanism is a theoretical as well as practical approach that was introduced to improve the traditional Seq2Seq architecture. [5, 48].

In a simple Seq2Seq architecture the entire source sentence is represented by the output of the final hidden state of the GRU (Figure 2.3a). This representation is called the sentence embedding. The decoder uses the sentence embedding to generate the target sentence. For source utterances with large sentence length; conditioning the target generation on the sentence embedding generates poor target utterances. The attention mechanism helps alleviate this issue by allowing the Seq2Seq model to focus on specific source utterance token at each generation timestep.

Figure 2.3b illustrates the attention mechanism proposed by Bahdanau et al. [5]. In the attention mechanism, the decoder generation is conditioned on the weighted combination of all input states, instead of the last state (Equation 2.3). The weight value denotes the importance of a certain encoder state output for a specific decoder generation timestep. If the input and output tokens are aligned then the α score will be high for those encoder states. For example, in the machine translation task certain English words have a one-to-one mapping with their German translation while certain words might consist of multiple correspondences.

$$a_{ij} = f(s_{i-1}, h_j) = v' \tanh(W_1 s_{i-1} + W_2 h_j) \tag{2.1}$$

$$\alpha_{ij} = softmax(a_{ij}) = \frac{\exp(a_{ij})}{\sum_j \exp(a_{ij})} \tag{2.2}$$

$$c_i = \sum_j \alpha_{ij} h_j \tag{2.3}$$

The attention scores α_{ij} are computed as a non-linear function of the hidden states, generating an attention parameter a_{ij} (Equation 2.1). The attention parameter $a_{i,j}$ is normalized using the softmax function to provide a probabilistic distribution across the input states at each decoding timestep j (Equation 2.2).

$$\begin{aligned} a_{ij} &= s_{i-1}^T h_j \ (dot - Luong) \\ &= s_{i-1}^T W_a h_j \ (general - Luong) \\ &= v^T \tanh(W_a[s_{i-1}, h_j]) \ (concat - Bahdanau) \end{aligned} \tag{2.4}$$

The alignment model which is used to score the mapping between the output tokens and the source hidden states can be proposed differently [48]. Equation 2.1 describes the attention model presented by Bahdanau et al. which is the additive/concat approach. Luong et al. [48] propose a multiplicative scoring model called the dot or general approach. We present the different scoring models in Equation 2.4. In all our affect responsive dialogue system models we use the additive alignment scoring model proposed by Bahdanau et al.

2.2 Dialogue System

Models that can generate utterances in response to an input utterance like dialogue system, conversational agent and chatbots have been a fairly researched problem studied in Artificial Intelligence as a variant of the Turing test problem [81, 15]. The research done in this domain is based on statistical language modeling, lexical/contextual rule based systems [87, 14], and, as a slot filling problem [7].

D.Jurafsky and JH Martin categorize chatbots in two primary categories: rule-based and corpus-based systems [38]. Rule based systems were some of the initial chatbots. ELIZA [87] and PARRY [14] were one of the first rule based conversational systems. ELIZA was designed to utilize and pattern user statements and to generate the response using directives provided in the form of 'scripts'. ELIZA was tested using a script that emulates a Rogerian psychotherapist. Modern chatbots like ALICE also utilize a lot of the features of ELIZA.

```
Sample Dialogue of ELIZA:
Men are all alike.
IN WHAT WAY
Theyre always bugging us about something or other.
CAN YOU THINK OF A SPECIFIC EXAMPLE
Well my boyfriend made me come here.
YOUR BOYFRIEND MADE YOU COME HERE
He says Im depressed much of the time.
I AM SORRY TO HEAR YOU ARE DEPRESSED
..
..
```

PARRY was designed to study schizophrenia. PARRY utilized conditioning on top of ELIZA like regular expressions. If the system detects 'anger' or 'fear' it can modify its output based on the intensity of the affect.

Corpus based chatbots can be categorized in two categories: Information Retrieval (IR) based and sequence-to-sequence trained chatbots. An IR based chatbot utilizes statistical techniques to perform some sort of ranking between a new user source utterance and a corpus of source-target utterance pairs, returning the target response utterance from the corpus that matches the new user utterance the best. Research suggests that there are two primary approaches in an IR based dialogue system:

- Return the target response utterance of the source utterance in the training corpus which has the highest similarity to the new user utterance.
- Return the target response utterance that best matches the new user utterance.

Some sort of similarity metric is used to rank and find the best match like the cosine similarity [36, 44]. There are various IR based chatting and question answering systems that are used in practice. Some examples are: Cleverbot[2] and Microsoft's 'XioaIce' system.

Recently there has been a lot of advancement in Sequence-to-Sequence (Seq2Seq) neural networks for dialogue systems inspired from the work in Neural Machine Translation [83]. Seq2Seq systems are generation models that utilize a language model trained using a corpus and generate a response word by word. There have been various modifications to the simple model so that the system is able to generate more diverse and contextual response that might make sense over a longer period of conversation.

Serban et al. propose a hierarchical model for dialogue response generation [71]. The model introduces an additional context RNN that keeps track of the context of the previous utterances in the dialogue. This allows the architecture to utilize much longer dialogue context to allow for much more responsive and engaging conversation. The authors also propose a variational approach for a hierarchical model [72]. The variational model makes the context representation much more diverse and allows for varied response generation for an input utterance. Xing et al. extend the model to include word and utterance level attention in the hierarchical model [88] to generate better responses. All these research approaches show that hierarchical models are able to generate contextually significant, longer and meaningful responses.

Some research tries to exploit adversarial learning techniques to make the latent representation invariant so that the system is able to generate longer conversations with meaningful replies as well [46].

[2] https://www.cleverbot.com/

2.3 Emotion Analysis

We use this section in continuation to Section 1.2.2 to explain the emotion analysis problem from a linguistic perspective. Even though we conduct experiments on both, sentiment and emotion, we only describe the emotion spectrum as that is our ultimate goal. The sentiment domain allows us to test our theories on a smaller scale to find out if the approach is working. Also, sentiment (positive and negative) is easily discriminative whereas our studies (Table 1.3 - Table 1.4) show that emotion is really difficult to distinguish between.

Emotion can be defined as 'A strong feeling deriving from one's circumstances, mood, or relationships with others.' or 'Instinctive or intuitive feeling as distinguished from reasoning or knowledge.'[3] Emotion analysis is an extensively studied field in behavioral psychology [20] as a result of which it has also been studied in computer science. Facial expression detection for emotion analysis has been studied in Human Computer Interaction [21] and in speech synthesis to automatically detect emotion from the tone [18]. Emotion has also been widely studied in computational linguistics in the field of opinion mining and affective computing. We use this chapter to briefly describe the emotion models and emotion annotated datasets available.

2.3.1 Emotion Models

There are various emotion models available today. Emotion models consist of a set of exclusive emotions called basic emotions. These emotions are referred to as primary emotions and they can not be represented as a combination of other emotions, and, we can use these emotions to represent other complex emotions. Atlas of Emotions[4] is an excellent example of how complex emotions can be derived from the basic emotions. Table 2.1 mentions some of the most frequently used models of basic emotions which we will explain briefly in this section.

P. Ekman, one of the primary researchers in the field of emotion detection utilized and created a vast database of facial expressions. He used these expressions to suggest a set of six universally recognized basic emotions. He later expanded his set of emotions by adding 12 new positive and negative emotions [23]. J. Russell [66] mentioned that all emotions could be categorized as different degrees of three basic bipolar dimensions: pleasure-displeasure, degree of arousal, and dominance-submissiveness. R. Plutchik [61] arranges emotions as a color wheel where the vertical dimension represents intensity and

[3] https://en.oxforddictionaries.com/definition/emotion
[4] http://atlasofemotions.org/#actions/

Model	Year	Emotions
Ekman	1972	anger, disgust, fear, joy, sadness, surprise
Russell	1977	pleasure/displeasure, degree of arousal, dominance/submissiveness
Plutchik	1986	anger, anticipation, disgust, fear, joy, sadness, surprise, trust
Shaver	1987	anger, fear, joy, love, sadness, surprise

Table 2.1: Different models of basic emotions.

the circle represents the degree of similarity between emotions (Figure 2.4). P. Shaver [73] represents emotions in a tree structure where the basic emotions are at the main branch and each branch has its own sub-categories.

2.3.2 Emotion Datasets

There have been various works that present multi-domain datasets which were human or automatically annotated with emotions. We use this section to present the details of some of the existing emotion annotated datasets. Even though there are some emotion lexicon datasets that present a list of affective words annotated with their corresponding emotion label, we will only focus on datasets which contain sentences annotated with their emotion. We describe the datasets that we've mentioned in Section 1.2.2.

International Survey On Emotion Antecedents And Reactions (ISEAR) was a popular project initiated in the 1990's by Klaus R. Scherer and Harald Wallbott [68]. In the project a large group of psychologists all over the world performed a survey with psychology and non-psychology students to assess different situations which would warrant an invoking of a specific emotion. The FairyTales dataset project was started by C. O. Alm et al. [2] In their project they were trying to present a distinctive emotion classifier trained on multiple different features which were extracted from the dataset they presented. In their dataset they perform manual annotations on a series of children's fairy tales. The NRCTEC dataset was compiled by Saif M. Mohammad [53] using Twitter as a data source. They selected six hashtags corresponding to each emotion in the Ekman emotion model (eg. #anger, #sadness, #joy etc.) and searched for Tweets that had any of those hashtags present. The assumption was that if a Tweet had any of those hashtag present it was a good indicator that the Tweet was reflecting that particular emotion. The BlogPosts dataset was created by Aman et al. [3] by gathering data from web blogs. The authors created a list of seed words similar to synonyms of the emotion classes and then looked for sentences from the blog post that contained any of those seed words. Manual annotators measured the degree of success of their data accumulation technique. The FairyTales and

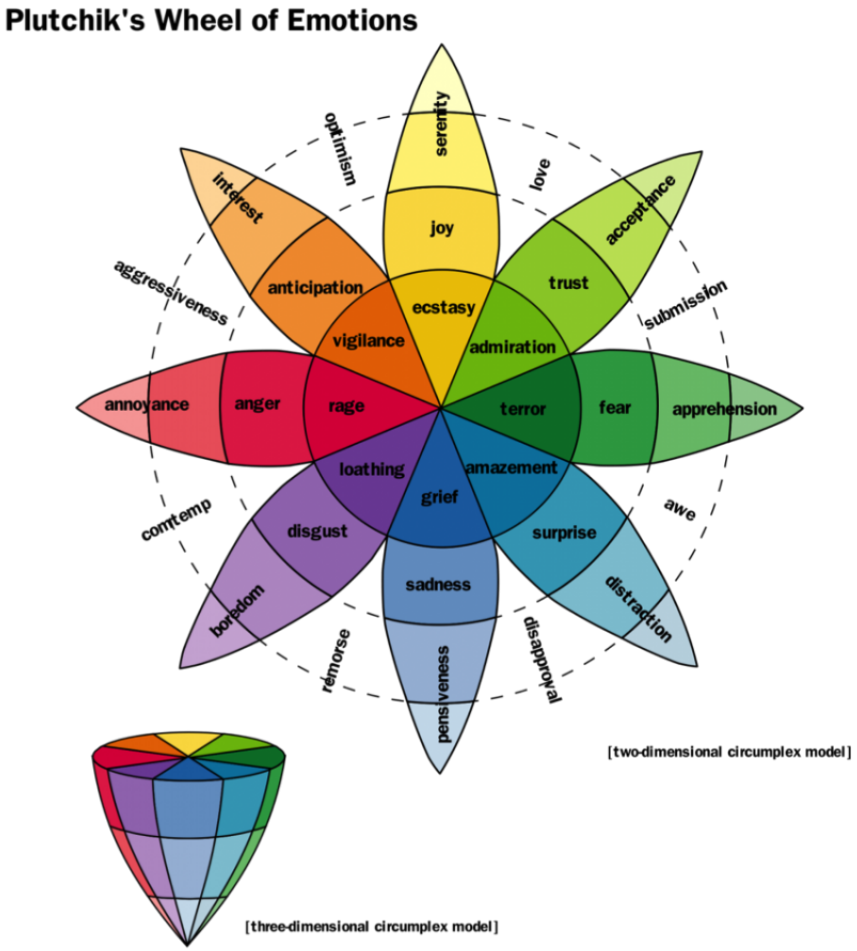


Figure 2.4: Plutchik's wheel model of emotion.

BlogPosts dataset is presented in two variations. The authors present all the data and a subset of that dataset which saw higher mutual annotator agreement. There are various other datasets that were gathered using Twitter but we only mention the readily available datasets which used manual annotation to verify the accuracy of their data annotation approach.

2.4 Emotion Classifier

Datasets described in the previous section along with the SemEval2007-Task14 (Affective Text) dataset are considered as the emotion benchmark datasets. The SemEval emotion dataset comprises a set of news headlines with a numerical rating on the 100-point scale for each emotion[5] [78]. These datasets have been extensively used to conduct emotion recognition and classification research in computer science. Most of those works are limited to training and prediction on a really small or fixed domain dataset. Our assessment (Figure 1.1) shows that classifiers trained on the following dataset are rarely able to predict emotion in text from cross-domain topics. We believe that Twitter is an excellent source for a general and cross-domain datasets. As a result we focus on some classification systems that were trained using a large amount of data gathered from Twitter, with the belief that they can recognize general emotion better. Twitter as a service provides a search API through which users can get a sampling of Tweets that have the passed query word present in it. Using this approach, most of the work that uses Twitter to gather the data set a list of seed words (often the emotion labels or their synonyms) as hashtags or individual words. Some work also utilizes emoticons or emojis to gather affective tweets.

Wenbo Wang et al. selected a list of hashtags which were synonyms of the emotion classes as seed words and gathered a set of 2,500,000 Tweets [85]. 400 Tweets were selected as a subset of the gathered Tweets and were manually annotated to verify the effectiveness of their data gathering process. The classifiers were built using features like n-grams, lexicon lists, part of speech (POS) tags and adjectives from the accumulated data and tested on the manually annotated dataset. Maryam Hasan et al. gathered a set of 124,000 Tweets using a set of hashtags and the Tweets were given to a group of psychologists and non-psychologists [31]. It was seen that the non-psychologist group had very poor mutual annotator agreement and, as a result, they conclude that crowd sourcing is not an effective technique to obtain emotion annotations. They further trained K-Nearest Neighbor and SVM classifiers using features like unigrams, negation, emoticon and punctuations. Roberts et al. gathered a set of 7000 Tweets containing emotion provoking topics/terms like 'Valentine's Day' and 'Christmas' [65]. Each Tweet was annotated with one or more emotions it contained and individual SVM classifiers were trained with different features. The NCRTEC dataset mentioned in the previous section was gathered by Mohammad et al., and contains about 21,000 Tweets that were selected using emotion hashtags [53]. A classifier was trained using unigram and bigram as features and the classifier was tested on the SemEval 2007-affective text dataset. Even though all these approaches are able to classify emotions to a certain degree of accuracy they posses certain drawbacks. These

[5] http://web.eecs.umich.edu/~mihalcea/affectivetext/

models are susceptible to having a very small training data set and hand curated features. This results in a good emotion prediction for a certain domain but poor performance across other datasets.

We would like to describe two approaches that utilize billions of Tweets and complex deep neural network architecture to train the classifier. DeepMoji is an architecture proposed by Felbo et al. where they select a set of 64 famous emojis and gather 56.6 billion Tweets which are processed and then filtered to 1.2 billion [25]. Their architecture consists of two bi-directional LSTMs [34] which is a type of RNN model (Section 2.1.3) with attention. Their model is able to predict the top five emojis that best describe a particular text sentence. They use their model to solve various text classification tasks by using transfer learning. In this approach they freeze all layers, but one, of the neural network architecture in the last stage and the unfrozen layer is trained further on task specific datasets like the ones mentioned in the previous section [24]. They only evaluate on a subset of the Ekman emotion model since their model did not see sufficient training data for the remaining classes. Colnerič et al. provide a neural network architecture especially for the task of emotion classification [17]. They use hashtags of emotion classes of Ekman, Plutchik and POMS [50] and gather 73 billion Tweets. The authors use an RNN and CNN neural network with dropout to create their classifier. Finally transfer learning is applied at the final layer and the softmax classifier for the different emotion models. We found that this emotion classification method works really well for our need to automatically annotate a huge corpus with the Ekman emotion labels.

2.5 Controlled Text Generation

There has been some work in the field of unconditional text generation like the work of Bowman et al. where they use a Variational Auto-Encoder (VAE) to generate sentences [9]. Variational Auto-Encoders (VAEs) [41] consist of encoder and decoder neural networks. The encoder converts the input data into its latent representation and the decoder generates text by sampling the latent representation obtained from the encoder. The authors show that by sampling two points from the latent representation and transitioning between them they could generate sentences that slowly move from one to the other. However this model generates random and uncontrollable sentences.

We use this section to focus on fairly recent techniques that utilize neural architecture to perform controlled text generation. Apart from these approaches there has been a lot of work that utilizes hand crafted linguistic rules and features along with syntactical grammar

rules that can generate controlled text. For a detailed review of some of those linguistic approaches the survey work by Gattet et al. [28] can be referred to.

Hu et al. introduced a conditional text generation model [35] using a VAE as the generator and encoder and attribute discriminators to distinguish between the task entities (sentiment or tense). Their model uses the wake-sleep algorithm [33] for conditional text generation. The wake step utilizes samples generated from the encoder network using the training data to update the decoder and the sleep step updates the encoder network with samples generated by the decoder. The authors use this model to generate sentences conditioned on user-specified attributes like sentiment and tense. The work by Rajeswar et al. use adversarial learning to generate controlled text [64]. They utilize a model proposed by Radford et al. [63] that uses deconvolution CNN in the GAN discriminator to perform unsupervised representation learning. The authors perform conditioning on the generator and discriminator by concatenating a feature vector filled with ones or zeros to denote the presence or absence of the conditioned attribute at the output of each convolution layer. This method was used to generate text conditioned on questions and sentiment. Wang et al. propose a modified GAN [29] architecture to generate sentiment-conditioned text [84]. They use multiple generators coupled with a single sentiment discriminator so that each generator can focus on generating text which has a single sentiment affect label. A similar type of model is presented by Yang et al. They present a modified semi-supervised VAE architecture that can be applied to the task of conditional text generation [90]. The authors present the reasoning that a simple VAE that uses LSTM in the encoder and decoder is not able to perform as well as an LSTM applied to the language modeling task. As a result, the authors replace the decoder LSTM with a dilated CNN. A dilated CNN can control the amount of prior input to include in the context required to generate the output at the current time-step. The authors utilize a semi-supervised VAE proposed by Kingma et al. [40] which incorporates discrete data label as additional variables during training. This modified semi-supervised VAE is used to generate text conditioned on sentiment (5 star - 1 star) rating of Yelp[6] reviews.

The task of controlled text generation has also been applied to different domains. Lebret et al. propose a model to generate sentences that can be used as the biography introduction of people [43]. They use the tokenized first sentence of the Wikipedia biography dataset[7,8] to build a conditional neural language model. They use a fact table (Knowledge Base) conditioned language model where the table is made up of key-value pairs. The key are special symbols generated by the language model which is replaced by the best value

[6] https://www.yelp.com/

[7] https://en.wikipedia.org/wiki/Wikipedia:WikiProject_Biography

[8] https://github.com/DavidGrangier/wikipedia-biography-dataset

derived from the facts table. Similarly, Peng et al. propose a model that can generate and control stories [59]. They propose an analyzer and generator model for the two tasks. In the storyline control task the analyzer extracts the keywords from the stories and for the story ending valence (happy or sad) task the analyzer is a classifier that predicts the stories valence. For both tasks the generator is a conditional language model where the conditioned attributes are the entities produced by the analyzer. For the ending valence control task the model generates story endings given the valence label and the story beginning. Whereas, for the storyline control task the model generates a story given a set of keywords.

2.6 Text Style Transfer

Text style transfer is a task where we want to convert a source sentence of a specific style into a sentence of another style while preserving the content of the original sentence. Text style transfer is an important task that can be used in various fields like help-support systems as it can allow the system to only generate compassionate, neutral and polite response to allow effective and longer user engagement.

Jhamtani et al. present an approach to transform modern English text to Shakespearian English using a Seq2Seq network along with a pointer network [37]. The Seq2Seq model generates words using a probability distribution over the vocabulary words whereas the pointer network provides a probability distribution over the input words to the Seq2Seq model so that the system can learn to copy words which are same between the source and destination styles. The model is enhanced by using a dictionary mapping between modern and Shakespearian English to augment the pre-trained embeddings with words that were not seen while training due to a limited training dataset.

This field also deals with the fact that sometimes we can not obtain parallel corpora of sentences in the two styles and hence we have to use different novel approaches to deal with that. Shen et al. propose a model which can transfer style by training on a non-parallel training corpus [74]. A non-parallel corpus means that during the training phase the model is not able to see the proper transformation of a source sentence into its target sentence. The authors use a technique called cross-alignment where they use a variational auto encoder and a discriminator in the architecture. The encoder is responsible for converting the source sentence into its latent representation and the generator uses that along with the style encoding to generate the sentence again. The authors assume that if this latent representation (which is style invariant due to the adversarial discriminator) along with the original source style information is able to reconstruct the source sentence then the latent representation along with the new style information will be able to construct the

target sentence with the required style while preserving the content of the source sentence. Similarly, Carlson et al. utilize a technique called zero-shot learning where the model does not see the actual transformation/conversion between the source and target sentences during training [10]. The authors use different version of the bible text and treat them as very effective style transformation between text pairs. The technique is adopted from machine translation systems and the authors state that the system can perform much better than a basic statistical machine translation system like Moses[9].

Fu et al. present two models to perform sentiment style transfer between text sentences [27]. The model uses two techniques of multiple decoders and style embedding to tackle the task of style transfer. At the basic level both models use an end-to-end trainable Seq2Seq model which consists of an encoder and a decoder. The style embedding approach has an additional embedding vector which is appended to the encoder representation of the source sentence and the decoder is responsible of generating varied transformed sentences based on the different emotion embedding appended to the encoder representation. The second approach uses multiple decoders in the architecture, one for each style. The idea is that each decoder will learn to generate sentences of a particular style or form. In both these approaches the model also contains an adversarial discriminator whose job is to make the encoder representation style- invariant so as to only save the content and allow the decoder to generate style specific sentences.

Zhang et al. propose a model called SHAPED (Shared-Private Encoder-Decoder) [94]. This model has shared parameters for both the encoder and decoder which are learned over the whole training dataset and private parameters for the encoder and decoder which are learned only from the corresponding style label items in the training set. The shared parameters are responsible for learning general language specific information like a language model and the private parameters learn specific information like style which might be different for different sentences. Li et al. propose an approach to modify a source sentence's style attribute to generate a transformed sentence [47]. The authors say that using adversarial learning to produce style invariant representations does not produce quality target sentences. They propose that an actual transformation only requires the changing of a few target attribute phrases while retaining the other words. Their approach learns a list of text attributes for each style. Then the model detects style specific text attributes in the source sentence and deletes those attributes. The text attributes for the target source are retrieved from the list and then a neural network generates grammatically proper sentences by merging the source sentence and the target specific style attributes.

Santos et al. use the task of style transfer to transform offensive textual sentences from

[9] http://www.statmt.org/moses/

websites like Twitter[10] and Reddit[11] [67] into non-offensive sentences. They use a modified collaborative classifier instead of the adversarial discriminator. They also introduce a new loss function called the cycle consistency loss. A new approach by Prabhumoye et al. performs style transfer by using an additional Seq2Seq model during end-to-end training to perform neural machine translation [62]. This model is similar to the Style-Embedding approach of Fu et al. [27] except the authors add a back-translation component before generating the invariant latent representation. The source sentence is translated into another language which is then passed through an encoder to generate the latent representation. The authors hypothesize that machine translation retains the content of the source sentence but is helpful in removing the specific style characteristics. Yang et al. design a model where the discriminator classifier is replaced by a language model trained on the target domain [89]. The authors state that the discriminator classifier is not an accurate method during training to determine whether the generator is generating proper style enforced and syntactically proper sentences. Hence they propose to train a language model on the target domain sentences and then evaluate the effectiveness of the generated sentences using the Cross Entropy loss which would give a proper and quantitative measure of the training process.

2.7 Attribute Conditioned Dialogue Systems

An attribute/entity conditioned dialogue system is a dialogue system where the decoder can generate distinct responses based on different attributes chosen by the user during run time. A model might be used to perform different actions or generate specific responses based on different attributes selected by the user. An attribute conditioned dialogue system incorporates two techniques described in the previous Sections - 2.5 and 2.6.

Li et al. propose a dialogue model conditioned on the speakers [45]. The authors propose two models namely the speaker and the speaker-addressee model. The speaker model incorporates an embedding for the speaker that is generating the response sentence. The speaker-addressee model incorporates the fact that each user might use a different style of speaking while talking to different people hence the embedding also incorporates the identities of who said the source sentence and who is saying the response sentence. These embeddings are trained in an end-to-end fashion during the training of the general model. Similarly Zhang et al. also present models that can learn information about the profiles of the addressee and the initial speaker [93]. They use various ranking and

[10] https://twitter.com/
[11] https://www.reddit.com/

generation techniques along with modified memory network to design a dialogue system. Similarly, Herzig et al. propose a model to provide customer support help by incorporating personality traits [32]. The personality traits are learned automatically during training.

Domain specific approaches like the approach proposed by Asghar et al. provide a model to generate affective responses by using three approaches, namely, affect word embeddings, affect-based modified cross-entropy loss and an affect diverse beam search decoder during generation [4]. Zhou et al. propose a similar model for the same task of emotional chatting [95] model which is basically a dialogue system that can generate varied emotional responses based on the emotion selected by the user during runtime. The model architecture uses three mechanisms, namely, emotion embeddings, internal RNN state for emotion and an external memory which consists of an emotion vocabulary. The authors work on a Chinese dataset to perform the task. Another model by Niu et al. uses a fusion model to generate polite replies [57]. They use a Seq2Seq model with attention trained on a conversation dataset and a language model trained on a corpus of polite utterances. By fusing the two approaches they are able to generate dialogue utterance responses which are polite in their affect.

Chapter 3

Methodology and System Description

We use this chapter to describe the dialogue dataset collection and annotation process as well as the system model.

3.1 Twitter Dialogue Dataset

Twitter is a micro-blogging platform where users can post short texts with a 280 character limit. The limit was increased from 140 to 280 characters in November, 2017. Twitter is an excellent source to acquire general cross-domain and topic datasets like we described in Section 1.2.3. Twitter offers fast and convenient APIs[1] to gather the tweets being posted by Twitter users.

3.1.1 Acquiring The Dialogue Dataset

Even though Twitter allows fast APIs for distinct data collection there is no API that can directly allow us to gather a dialogue dataset. Instead we make use of two API endpoints provided by Twitter:

- **statuses/lookup** The statuses/lookup API endpoint takes in a list of tweet ids (maximum of 100) and returns a JSON object with all information of that tweet.

[1] https://en.wikipedia.org/wiki/Application_programming_interface

- **statuses/filter** The statuses/filter API endpoint allows users to set filters on the public Twitter feed which provides a random sampling of the Tweets being posted on Twitter by users in real-time. The filters can have constraints like the keywords (up to 400) present in Tweets, user accounts (up to 5000 users) publishing the Tweets or the location of the Tweets.

In our approach we use the filter API endpoint with the language restricted to English and the location constrained to the state of California, USA with the assumption that this would generate a large dataset due to the size of Twitter user base in California[2,3]. The filter API constantly provides us with a set of Tweets that match our constraints in the form of JSON objects. The JSON object has various key value pairs. The keys we use are called `id` which provides a unique numerical ID for the Tweet, `text` which contains the actual text of the Tweet and `in_reply_to_status_id_str` which allows us to check if the Tweet is in response to any Tweet or if it is an original Tweet. Using this approach we can check if we have dialogue tweets present in the Tweets obtained by the filter API. If the original Tweet is not present we use the lookup API endpoint to request the Tweet mentioned in the `in_reply_to_status_id_str` key. Using this approach and backtracking on the set of Tweets obtained by the filter API we gathered our Twitter dataset.

Using our approach we can create a dataset in which there are multiple utterance turns. However, for our task we only need a dataset which contains dialogue pairs. Hence, we use the dataset gathered to create utterance pairs. The statistics about the original and the pair dataset are given in Table 3.1. We randomly select some utterance pair dialogues to use during the training of our model (Table 3.2).

Dialogue Dataset	Dataset Count
Multi-turn dataset	1,368,102
Utterance-pair dataset	2,199,366

Table 3.1: Twitter Dialogue Dataset Count

[2] http://tweeplers.com/cities/?cc=US

[3] https://www.allbusiness.com/twitter-ranking-which-states-twitter-the-most-12329567-1.html

Dataset	Dataset Statistics		
	Training	Validation	Testing
Utterance-pair dataset	400,000	5,000	5,000

Table 3.2: Model Training Dataset Statistics

3.2 Emotion Annotated Dialogue Dataset

We utilize the emotion classifier model proposed by [17] which is described in Section 2.4. The authors utilize hashtags keywords for the various different emotion models and train a hybrid RNN and CNN model with transfer learning on the 73 billion collected Tweets. This models was able to allow us to annotate our Twitter dialogue dataset with the appropriate emotion labels in the Ekman emotion model [21]. The Ekman emotion model consists of six emotion labels - anger, disgust, fear, joy, sadness and surprise.

Even though we work on the emotion domain, we also utilize the emotion labels provided by the pre-trained emotion classifier to label the Twitter dialogue dataset with the corresponding sentiment label (positive or negative). We find that the emotion labels could be transfered to sentiment labels as they are a more fine-grained version of the sentiment domain. Table 3.3 shows the corresponding mapping between the emotion and the sentiment labels.

Sentiment Labels	Emotion Labels
Positive	Joy and Surprise (+)
Negative	Anger, Disgust, Fear, Sadness and Surprise (-)

Table 3.3: Sentiment and Emotion Label Mapping

We find that the emotion label *surprise* can represent a positive as well as negative sentiment. In the dialogue utterance where the emotion label is *surprise* we consult the confidence value provided by the pre-trained classifier and choose the second highest probability class for reference. For instance, if the emotion suggested is surprise, we check the second highest emotion label suggested by the classifier. If the second highest probability confidence value lies in the positive set then the dialogue is marked as positive, otherwise it is labeled as negative. Thus we are able to use the emotion labels provided by the classifier to annotate our Twitter dialogue dataset with both emotion and sentiment domain labels.

3.3 Classifier Model

We use this section to describe, in detail, the emotion classifier used by us to annotate our Twitter dialogue dataset (Section 3.1) and to test the model transfer strength (Section 4.1). In our work we rely on the classifier model designed by Colneriĉ at al. [17] to provide us with the Ekman emotion label (Section 2.3.1) for a sentence.

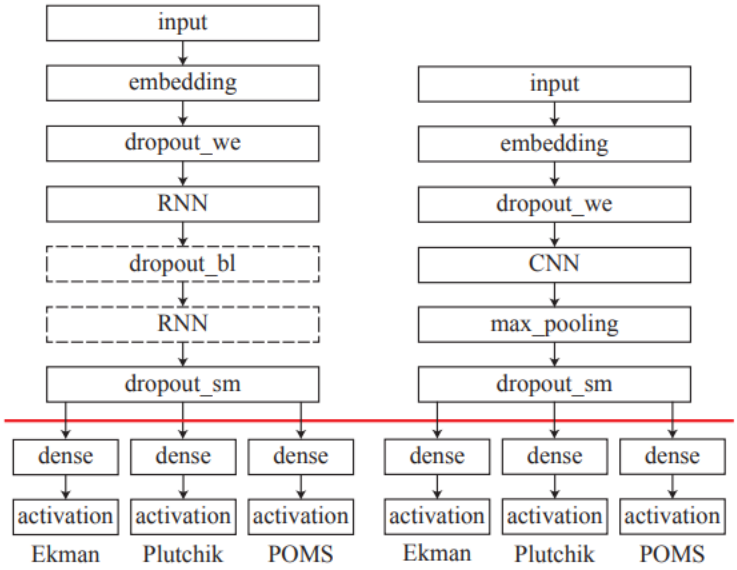


Figure 3.1: Classifier model proposed by Colneriĉ at al. [17] using unison learning.

Colneriĉ et al. train a classifier that can provide emotion labels for the Ekman, Plutchik and POMS (profile of mood states) emotion model. The authors use a collection of Tweets which contain the emotion labels or their synonyms as hashtags. Using these Tweets the authors train an RNN and CNN classifier as shown in Figure 3.1. The authors further propose the concept of unison learning. In this approach the classifier can be broken down into two phases. The first phase is where the classifier tries to learn the sentence representation using an RNN or CNN neural network model. The authors propose that sharing the classifier parameters till the sentence representation stage for all emotion models might allow the classifier to learn a better and more general emotion representation. The second phase contains a separate softmax layer for each emotion model. This layer is trained to predict the emotion label for a particular emotion model from the sentence representation. The authors say that this architecture learns a low-dimensional embedding that is informative enough for predicting all three emotion model categories at once.

In our preliminary experiments we test the classifiers ability to detect emotion and sentiment on the NRCTEC and sentiment140 Twitter dataset, respectively. We present the classification results in Section 4.3.

3.4 System Description and Methodology

In our work we propose an end-to-end trainable Seq2Seq model (Section 2.1.4) inspired by the model proposed by Fu et al. [27] for sentiment style transfer of a sentence in an auto-encoding task setting. They propose an adversarial learning component on the sentence representation produced by the encoder and use two distinct decoder models to reconstruct the sentence with sentiment style transfer.

We use the approach in a Seq2Seq affect (emotion or sentiment) conditional dialogue generation task. The adversarial component is applied on the latent representation of the source utterance and the two distinct approaches for the decoder model are used to design an affect (sentiment or emotion) responsive dialogue system. For the first approach we condition the decoder on the affect style embeddings and in the second approach we use multiple decoders - one for each distinct affect label.

The Seq2Seq model is augmented with an adversarial learning discriminator. This adversarial component's objective is to make the initial user utterance provided to the model affect-invariant by removing the emotional valence (style) from the source utterance while retaining the content of the utterance. We hypothesize that removing the style (affect) from the encoder generated latent representation of the source utterance retaining only the content information will allow our decoder to better utilize and condition the generated response on the user selected affect label. Otherwise, we suggest that if the source utterance is heavily affective and represents another affect than the one provided by the user for response generation it might bias the response generation process thus ignoring the user selected label. Also, if the source and target dialogue utterance heavily represent the same affect label, that might bias the generation process as well. Hence, we feel that by making the source utterance style-invariant we can allow different decoder approaches to better generate the affective response to the source utterance.

Figure 3.2 shows the complete process followed by our approach. The step-A in the model describes the approach mentioned in Section 3.1, 3.2. We accumulate our Twitter dialogue dataset $D = \{d_1, d_2, ..., d_n\}$, where each dialogue pair $d_i = (u_{i,1}, u_{i,2})$ with $u_{i,1}$ as the initial source utterance and $u_{i,2}$ as the target response utterance. This dialogue dataset is passed through the classifier (C) defined and proposed by [17]. The classifier provides

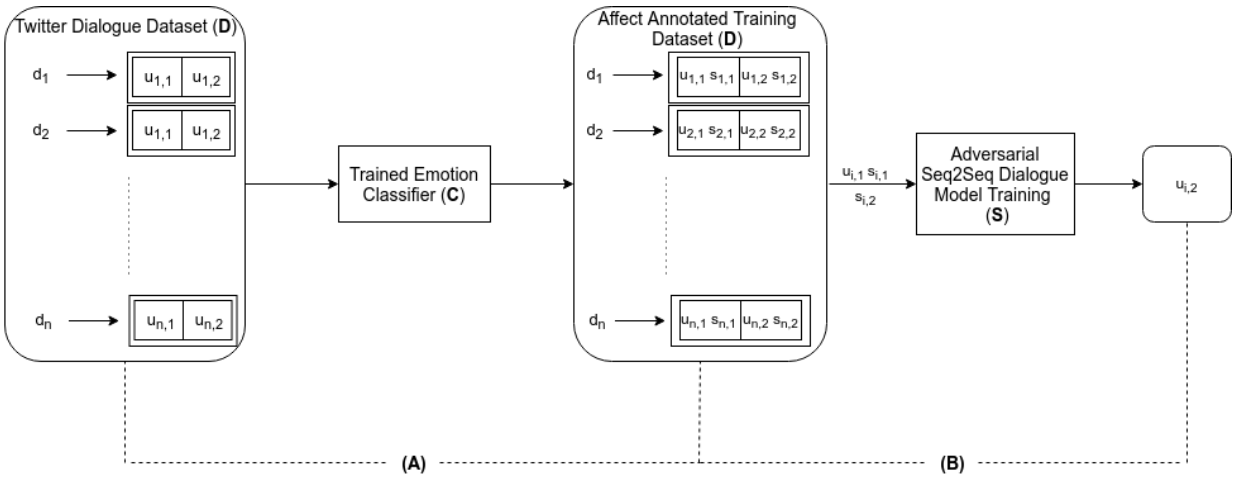


Figure 3.2: Overall Training Process

us with the affect label for each utterance in d_i. This provides us with our training dataset D where each training data point $d_i = (u_{i,1}, s_{i,1}, u_{i,2}, s_{i,2})$:

$$d_i = \begin{cases} u_{i,1}: & \text{the source utterance.} \\ s_{i,1}: & \text{the source utterance affect label.} \\ u_{i,2}: & \text{the target utterance.} \\ s_{i,2}: & \text{the target utterance affect label.} \end{cases} \tag{3.1}$$

Below we describe step-B which is the training process of our model. During the training process we use our dataset D to train the Seq2Seq model. At any given time-step/instance of training d_i is used. The first utterance, it's affect label and the affect label of the target utterance $(u_{i,1}, s_{i,1}, s_{i,2})$ is passed through our model which in turns tries to generate the target utterance$(u_{i,2})$. Figure 3.3, 3.4 describe the two training methods we use. Our method contains the encoder, decoder and discriminator.

3.4.1 Encoder

The encoder model is same for both decoder models. The encoder provides the latent representation of the source utterance $x_i = u_{i,1}$. The encoder contains a word embedding layer E_x that provides a high-dimension vector representation of each word in the training dataset D vocabulary in a continuous vector space (Section 2.1.2) and an RNN model (Section 2.1.3) called Gated Recurrent Unit (GRU)[19].

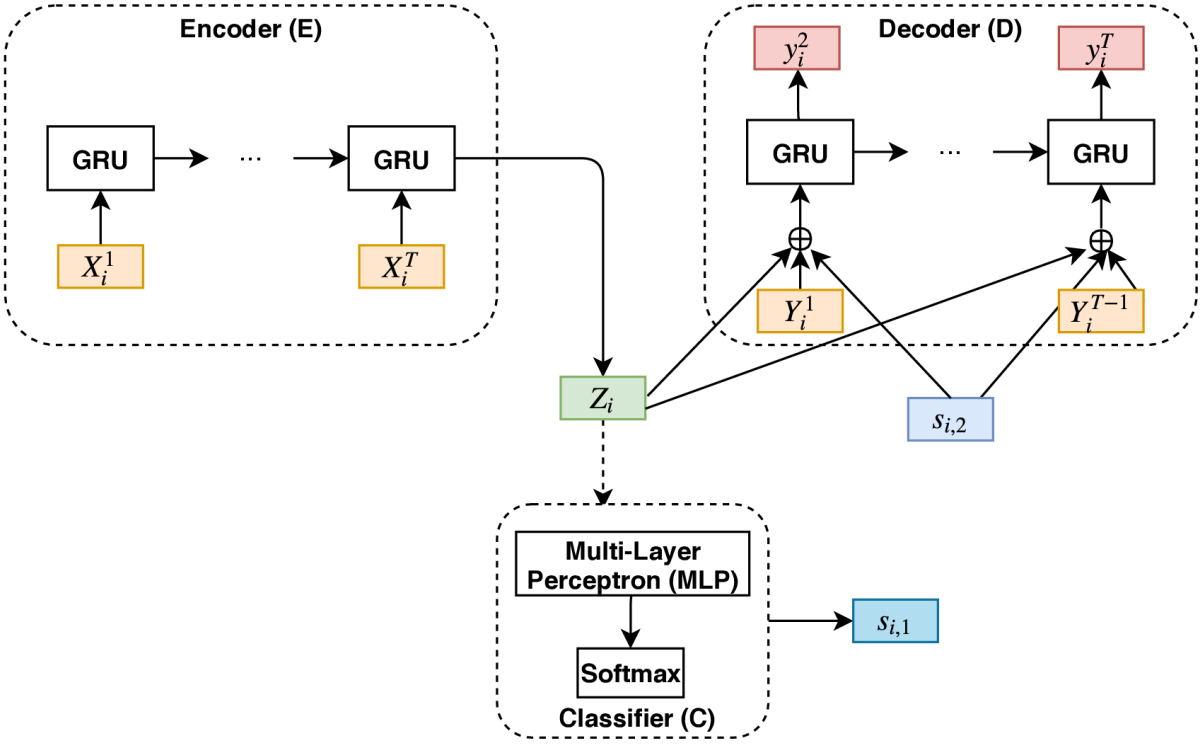


Figure 3.3: Training Model with the Style-Embedding Decoder

$$
\begin{aligned}
x_i &= u_{i,1} \\
x_i &= \{x_i^1, x_i^2, ..., x_i^T\} \\
X_i^j &= E_x(x_i^j) \mid X \in \mathbb{R}^{s \times Dim} \\
X_i &= \{X_i^1, X_i^2, ..., X_i^T\}
\end{aligned}
\tag{3.2}
$$

x represents the input to the training model; it consists of the source utterances. Each utterance x_i consists of a set of word tokens x_i^j. Each word has a distinct representation mapping in the embedding layer E_x denoted by X. The encoder GRU obtains the vector representation X^i of each word in the source utterance at each training time-step/iteration i (Equation 3.2). The encoder provides us with the latent representation of the source utterance, which is denoted by Z (Equation 3.3). We encapsulate all trainable parameters of the encoder, which consists of the weights of the GRU and the encoder-embedding layer, and represent them by Θ_e. Hence, the encoder model process can be described as:

$$
Z = Encoder(x; \Theta_e) \tag{3.3}
$$

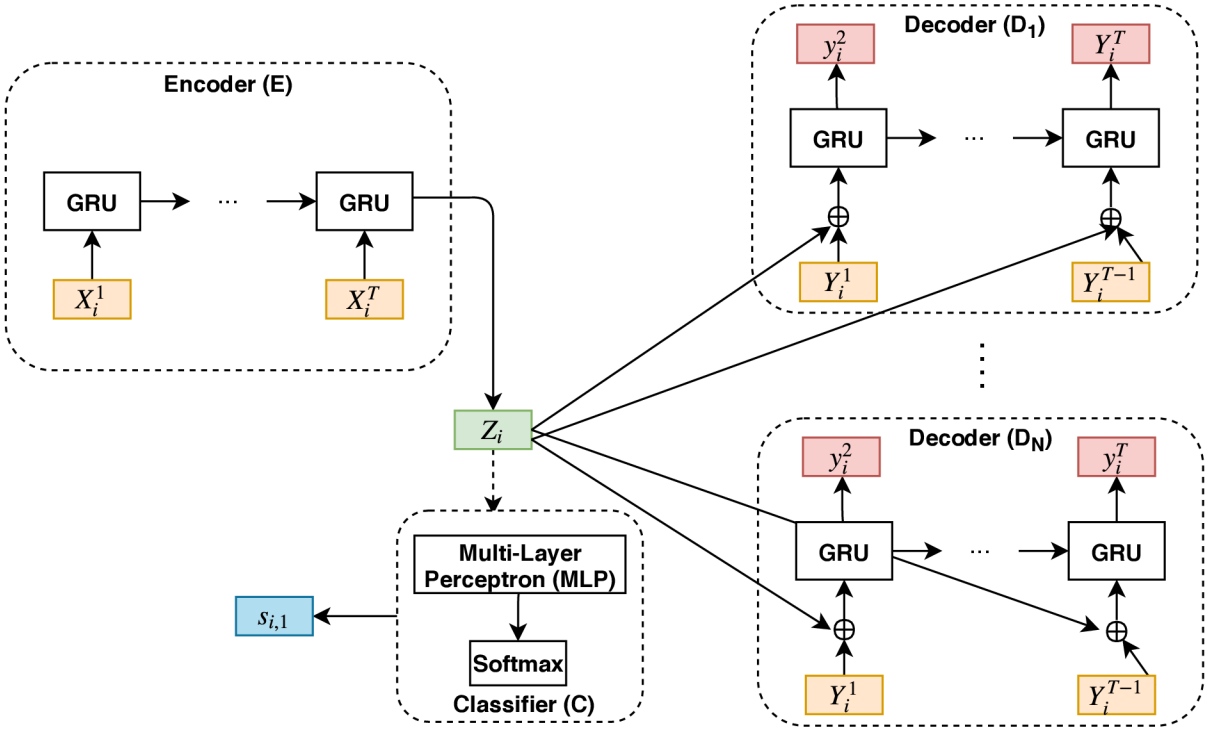


Figure 3.4: Training Model with the Multi-Decoder

3.4.2 Discriminator

The adversarial component in our model is the discriminator/classifier that ensures that the encoder does not learn the affective style of the source utterance. The model obtains the final hidden state of the encoder Z which represents the latent representation of the source utterance. This representation Z is passed through a dropout regularization technique [77] and the ReLU activation function [56] both of which allow the classifier model to avoid over-fitting on the latent representation. The output is then processed through a neural network linear layer (Section 2.1.1) and finally the LogSoftmax[4] function that allows the classifier to provide its final result that can be passed to the NLL-loss function[5].

[4] https://www.tensorflow.org/api_docs/python/tf/nn/log_softmax
[5] https://pytorch.org/docs/stable/nn.html

3.4.3 Decoder

The decoder is responsible for generating the target utterance from the source utterance's latent representation provided by the encoder as well as other attributes on which the response might be conditioned. The decoder consists of an embedding layer, GRU recurrent neural network and a linear layer that allows for the response generation.

$$\begin{aligned} y_i &= u_{i,2} \\ y_i &= \{y_i^1, y_i^2, ..., y_i^T\} \\ Y_i^j &= E_y(y_i^j) \mid Y \in \mathbb{R}^{s \times Dim} \\ Y_i &= \{Y_i^1, Y_i^2, ..., Y_i^T\} \end{aligned} \tag{3.4}$$

y represents the target utterance that the decoder tries to generate. We utilize a technique called teacher-forcing or scheduled sampling to generate the responses at training time [6]. In this approach at each time step i we randomly select either the decoder generated token at the $i-1$ time step or the ground truth target y_i which helps the decoder to learn the generation process more quickly. We utilize two distinct approaches for training which consist of different decoders, namely, style-embedding model and multi-decoder model.

Style-Embedding Decoder Model

Figure 3.3 describes the model that utilizes the style-embedding decoder for training. In this model we condition the generation process on an additional vector SE which is an embedding layer that provides a vector representation mapping for each distinct affect label in the training set. We condition the generation process on the schedule sampled token, affect invariant source utterance representation Z and the affect style vector that is obtained from the affect embedding layer which is trained in the decoder during the end-to-end model training phase, and, the hypothesis is that the affect invariant source utterance representation can encapsulate the content of the utterance whereas the affect embedding can learn the affect of the desired source utterance thus allowing us to condition the target generation on the desired affect.

Multi-Decoder Model

Figure 3.4 describes the multi-decoder model training process. The multi-decoder model does not condition the generated output on the style embedding. In this model we create

a separate decoder for each affect, which is two for sentiment and six for emotion. The encoder obtains the source utterance and provides us with the utterance vector representation which is provided to the decoder responsible for generating the target utterance in its desired affect. This way we condition the language generation on the source utterance latent representation and the schedule sampled ground truth token (during training) or token predicted at the previous step (during inference). By separating the decoder for each affect we hypothesize that the respective decoder learns the affect specific vocabulary and generation.

3.5 Training Process

We use this section to describe the training process of our approach as well as the loss function that we optimize for training. Θ_e, Θ_d and Θ_c denote the trainable parameters of the Encoder, Decoder and Discriminator (Classifier) respectively.

As seen from Figure 3.3 initially the source utterance x is passed to the encoder which provides the latent representation Z of the source utterance which is the final hidden state of the encoder GRU network (Equation 3.5).

$$Z_i = Encoder(x_i; \Theta_e) \tag{3.5}$$

This representation is passed to the adversarial discriminator (C) which tries to optimize the encoder and the discriminator classifier based on two distinct loss functions.

We use the classifier result to update the discriminator by minimizing the cross-entropy loss (negative log-likelihood of the LogSoftmax probability) as shown in Equation 3.6 where M denotes the training data size and l_i is the label selected for the passed Z_i (Equation 3.5).

$$L_{disc1}(\Theta_c) = -\sum_{i=1}^{M} \log\, p(l_i | Encoder(x_i; \Theta_e); \Theta_c) \tag{3.6}$$

The encoder is optimized by maximizing the entropy loss (minimize the negative entropy) over the classifier result as shown in Equation 3.7, where $H(p) = -\sum_i p_i \log p_i$ is an entropy of distribution p, M denotes the training data size and N is the number of affect labels. This ensures that the encoder is sufficiently confused and cannot truly choose the correct affect label for the source utterance [11]. The affect-invariant representation of the source utterance is passed to the decoder which generates the sequence tokens based on the language model defined by Equation 3.8 where the current token y_i^j is conditioned on

the Encoder output for source utterance x_i (Equation 3.5) and all previously generated target tokens $y_i^1, ..., y_i^{j-1}$.

$$L_{disc2}(\Theta_e) = -\sum_{i=1}^{M}\sum_{j=1}^{N} H(p\,(j|Encoder(x_i;\Theta_e);\Theta_c)) \tag{3.7}$$

$$P(y_i|x_i;\Theta_d) = \prod_{j=1}^{T_y} p(y_i^j|Encoder(x_i;\Theta_e), y_i^1, ..., y_i^{j-1};\Theta_d) \tag{3.8}$$

The style embedding decoder (Figure 3.3) obtains the affect invariant source utterance representation along with the affect label for the target utterance. The target affect label provides us with the embedding for the affect label. The affect embedding and the source utterance representation are used to condition the response generation process. Equation 3.9 describes the reconstruction loss (L_{gen1})and the total end-to-end loss ($L_{totalSE}$) when using the style-embedding model for training. The generation loss is conditioned on an extra variable parameter **SE** which denotes the affect style embedding layer. We minimize the cross-entropy loss between the generated and actual target tokens and use this loss to update the encoder and decoder models.

$$\begin{aligned} L_{gen1}(\Theta_e, \Theta_d, \mathbf{SE}) &= -\sum_{i=1}^{M} \log P(y_i|x_i;\Theta_e,\Theta_d) \\ L_{totalSE}(\Theta_e, \Theta_d, \Theta_c, \mathbf{SE}) &= L_{gen1}(\Theta_e, \Theta_d, \mathbf{SE}) + L_{disc1}(\Theta_c) + L_{disc2}(\Theta_e) \end{aligned} \tag{3.9}$$

The multi-decoder model (Figure 3.4) takes the source utterance representation and passes it to the decoder which is responsible for generating the target utterance affect label responses. Hence, the response can be conditioned on the source utterance representation and the scheduled sampled token. Equation 3.10 describes the generation loss L_{gen2} which is the cumulative average of the cross-entropy reconstruction loss encountered by each affect style decoder and the total end-to-end loss $L_{totalMD}$ of the model which uses multiple decoders.

$$\begin{aligned} L_{recon}(\Theta_e, \Theta_d) &= -\sum_{i=1}^{M} \log P(y_i|x_i;\Theta_e,\Theta_d) \\ L_{gen2}(\Theta_e, \Theta_d) &= \sum_{i=1}^{L} L_{recon}^i(\Theta_e, \Theta_d^i) \\ L_{totalMD}(\Theta_e, \Theta_d, \Theta_c) &= L_{gen2}(\Theta_e, \Theta_d) + L_{disc1}(\Theta_c) + L_{disc2}(\Theta_e) \end{aligned} \tag{3.10}$$

3.6 Hyperparameter Estimation

We use the Adam optimizer [39] with a learning rate of 0.001 and mini-batch size of 128 to update each individual sub-models (Encoder, Decoder and Discriminator). We run the training module for 100 epochs. The Cross-Entropy Loss[6] is used for the reconstruction and discriminator objective while the Entropy loss is used for the adversarial component (Equation 3.7).

- **Word Embedding** $200-dimension$ embedding trained on our Twitter Dialogue Dataset (Section 3.1) using Word2Vec [51] run for $50-iterations$ over the training dataset.
- **Encoder** A $1-layer$ Bi-Directional GRU [19] (Section 2.1.3) (200-dimension hidden size).
- **Decoder** A $1-layer$, 1-Direction GRU (200-dimension hidden size) with Bahdanau Attention [5].

All models use the dropout regularization [77] ($probability = 0.1$) in the latent representation to avoid over-fitting.

[6] https://pytorch.org/docs/stable/nn.html#torch.nn.CrossEntropyLoss

Chapter 4

Results and Analysis

We use this chapter to describe the evaluation metrics used to measure the effectiveness of our systems and the evaluation results obtained by our model on the affect (emotion and sentiment) responsive dialogue generation system.

4.1 Evaluation Metrics

We measure the effectiveness of our system using four evaluation metrics - Transfer Strength, Content Preservation, Word Overlap and BLEU [58]. For an Experiment Evaluation (E_{eval}) the model data input and output is defined in Equation 4.1:

$$E_{eval} = \begin{cases} x_i : & i^{th} \text{ source utterance.} \\ y_i : & i^{th} \text{ target utterance.} \\ s_i : & i^{th} \text{ target utterance actual affect label.} \\ s'_j : & \text{user selected generation affect label for } x_j. \\ y'_{i,j} : & \text{model generated response for } x_i \text{ and } s'_j. \\ X = \{x_i\} : & \text{set of source utterance}(1 \leq i \leq N). \\ S = \{s'_j\} : & \text{set of affect labels}(1 < i \leq M). \\ Y = \{y'_{i,j}\} : & \text{set of generated responses}(1 \leq i \leq N; 1 < j \leq M). \end{cases} \tag{4.1}$$

4.1.1 Transfer Strength

Transfer strength is a metric that is used to show the effectiveness of the dialogue system to transfer the affect style. The transfer strength is calculated using the pre-trained classifier [17] which we used to annotate our Twitter Dialogue Dataset (Section 3.2).

The classifier (C) takes as input $y'_{i,j}$ (Equation 4.1) and the prediction result is used to see if the generated sentence y' exhibits the affect selected by the user (s'_j) for generation. The classifier provides a confidence (probability) level score $C(y'_{i,j})$ for the target response generated by the model using the user selected affect label s'_j (Equation 4.1).

$$pred_{i,j} = \begin{cases} 0: & \arg\max_{s'_j} C(y'_{i,k}, s'_j) \neq s'_k;\ 1 \leq j \leq M \\ 1: & \arg\max_{s'_j} C(y'_{i,k}, s'_j) = s'_k;\ 1 \leq j \leq M \end{cases} \tag{4.2}$$

$$\begin{aligned} TS_j = TS(s'_j) &= \frac{\sum_{i=1}^{N} pred_{i,j}}{N} \\ TS_{model} &= \frac{\sum_{i=1}^{M} TS_i}{M} \end{aligned} \tag{4.3}$$

The classifier has a softmax function for the M-affect labels which takes an M-dimensional vector of arbitrary real values and produces another M-dimensional vector with real values in the range $(0, 1)$ that add up to 1.0.

The $pred_{i,j}$ is used to calculate whether the style transfer was successful or not. It is assigned a score of 1 or 0 based on the result obtained by passing $y'_{i,k}$ to the classifier C (Equation 4.2). $y'_{i,k}$ is the utterance generated by the model in response to the source utterance x_i and the user selected affect-label s'_k to condition the model response generation. The classifier (C) provides us a confidence score for the M-affect labels. We check if the predicted affect label with the highest confidence score is s'_k in which case $pred_{i,j}$ is given the score of 1 for the generated response, otherwise it is assigned a score of 0 (Equation 4.2).

TS_j is the Transfer Strength score of a model to generate the utterances for the user selected affect label $s'_j \in S$ (Equation 4.3). The overall score of the model that represents the capability of transferring all of the individual affect labels M (which is two for sentiment and six for emotion) is defined as TS_{model}, and is the mean of the scores of all individual $TS_j \mid 1 \leq j \leq M$ (Equation 4.3).

4.1.2 Content Preservation

Content preservation is an evaluation metric that ensures that the model generated utterance retains the context of the target utterance while effectively changing the affect of the model generated response utterance [27].

The content similarity metric is calculated as the cosine similarity metric (Equation 4.5) between the vector representation of the target utterance sentence y_i and the model generated sentence $y'_{i,j}$ (Equation 4.1). The vector representation of an utterance sentence consisting of words $\{w_1, ..., w_n\}$ is calculated by the $min, mean, max$ vector representation concatenation. The word2vec [51] representation of the words trained on the training dataset is used for calculating the sentence representation of an utterance sentence (Equation 4.4).

$$\begin{aligned} v_{min}[i] &= \min\{w_1[i], ..., w_n[i]\} \\ v_{mean}[i] &= \text{mean}\{w_1[i], ..., w_n[i]\} \\ v_{max}[i] &= \max\{w_1[i], ..., w_n[i]\} \\ v &= [v_{min}, v_{mean}, v_{max}] \end{aligned} \tag{4.4}$$

$$score_{i,j} = \frac{v_i \cdot v_j}{||v_i|| \cdot ||v_j||} \tag{4.5}$$

$$score^i = \frac{\sum_{j=1}^{M} score_{y_i, y'_{i,j}}}{M} \tag{4.6}$$

$$score_{total} = \frac{\sum_{i=1}^{N_{test}} score^i}{N_{test}} \tag{4.7}$$

The emotional words from a sentence are removed before calculating the vector representation of the target utterance and the model generated sentences. We utilize the emotion word list provided in the NRC Emotion Lexicon list [54].

The similarity score for each $y_i \in Y - score^i$ is calculated as the mean of the similarity score between y_i and $y'_{i,j} \,|\, j \in [1, M]$ where M is the number of affect labels for the dialogue system (Equation 4.6). $M = 2$ for the sentiment responsive dialogue system and $M = 6$ for the emotion responsive dialogue system. The total similarity score $score_{total}$ is defined as the mean of the similarity score of each $score^i \,|\, i \in [1, N_{test}]$ (Equation 4.7).

4.1.3 Word Overlap

Word overlap is a metric which shows the number of words which are common to both the target utterance y_i and the model generated response $y'_{i,j}$

$$\begin{aligned} sent_i &= set\{w_{i,1}, ..., w_{i,n}\};\ |sent_i| = n \\ WO_{i,j} &= sent_i \cap sent_j \end{aligned} \tag{4.8}$$

$$WO^i = \frac{\sum_{j=1}^{M} WO_{y_i, y'_{i,j}}}{M} \tag{4.9}$$

$$WO_{total} = \frac{\sum_{i=1}^{N_{test}} WO^i}{N_{test}} \tag{4.10}$$

If a sentence $sent_i$ consists of the words $\{w_{i,1}, ..., w_{i,n}\}$ then the word overlap between two sentences is defined as the words which are common between the two sentences (Equation 4.8). The word overlap score for a test sentence is defined as the mean of word overlap scores for each model generated sentence based on the user selected affect label (Equation 4.9). The word overlap score of a model is defined as the mean of the word overlap scores of each sentence in the test set (Equation 4.10).

4.1.4 BLEU score

We use the BLEU score (BLEU-1,2,3,4) [58] metric over the validation set evaluation after each epoch to decide which epoch test scores to consider for the final result presentation. BLEU score is a metric to check how well two sentences compare to each other in terms of similarity. We calculate the BLEU score between the actual target response y_i with the affect label s_i and the model generated response for the actual target affect label $y'_{i,j}$ where $s'_j = s_i$ (Equation 4.1).

4.2 Preliminary Result on Auto-encoding

In our approach we use the model inspired by Fu et al. [27] for the autoencoding style transfer problem. In their approach the authors try to transfer the sentiment of a source

sentence by using the style embedding and multi-decoder model augmented with the adversarial training component to make the encoder source sentence representation sentiment affect invariant. The work done by Fu et al. was done on Amazon reviews for the sentiment style transfer problem. Even though amazon reviews are similar to the Twitter tweets as both of them have informal language they differ as Amazon reviews offer product related reviews whereas tweets are often general and open for varied subjective interpretation in terms of the affect they convey (not always in response to any particular situation).

Therefore we ran the auto-encoding model on a Twitter sentiment dataset to make sure that style-transfer was possible on the Twitter domain before moving to the Seq2Seq domain to construct a dialogue system. We also tried variations in the model loss calculation to see if it would allow for better sentiment transfer. In this section we show the result obtained by us on sentiment transfer problem on the Twitter sentiment annotated tweets acquired by Sentiment140[1]. The tweets were acquired and annotated using distant supervision approach. Twitter **statuses/filter** API (Section 3.1) was used to look for tweets containing emoticons[2] and the tweets with emoticons representing a positive sentiment were labeled as positive, while the tweets containing emoticons that represent a negative sentiment were labeled as negative. During training we change the sentiment of positive tweets to negative and vice-versa.

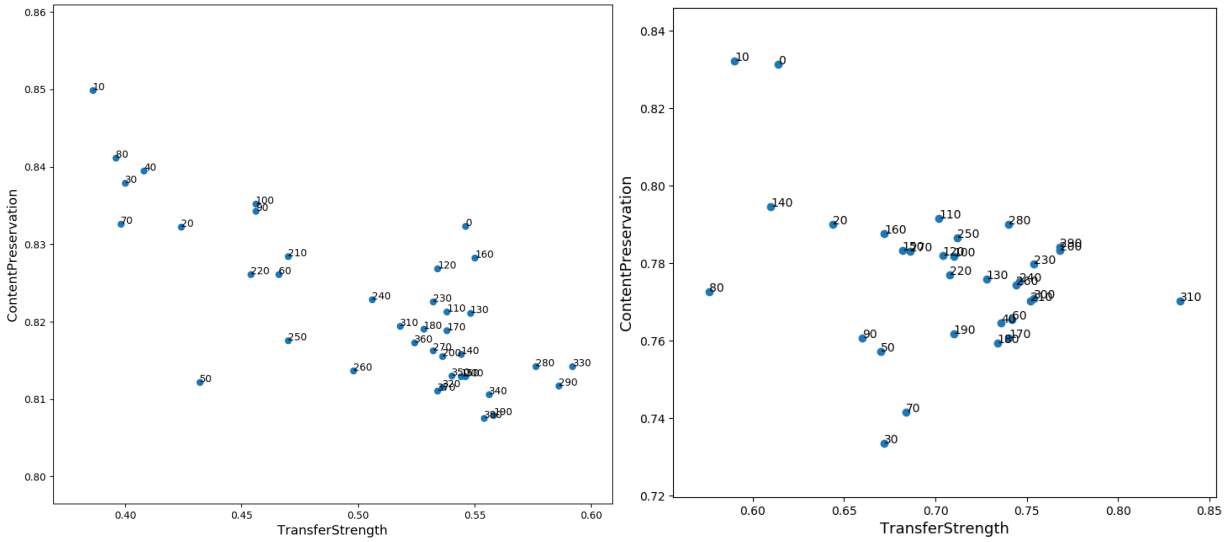
Figure 4.1 shows the TransferStrength vs ContentPreservation results obtained by the style-embedding and the multi-decoder model with an adversarial component on the auto-encoding Twitter sentiment dataset. Each point in the plot is annotated with the corresponding epoch which gives that result. We also provide some generated samples with the opposite sentiment for the epoch which have good TransferStrength vs ContentPreservation scores in Table 4.1. From these results we can observe that the model is able to transfer the sentiment effectively though the content is not preserved completely.

Inspired by the work done by Lample et al. [42] we introduce a slowly increasing lambda component on the adversarial loss component of the total loss of the model (Equation 3.9, 3.10). The lambda is slowly increased as the iterations increase with the hypothesis that the initial lower lambda would allow the model to learn the reconstruction effectively and quickly and later on the higher lambda will ensure that the encoder representation can learn to produce invariant latent representation of the source sentence (Equation 4.11). Unfortunately, in the experimental result we did not find the lambda to have a smooth transition effect on the textual data as it was seen on the image reconstruction and style transfer task. Hence, we do not use the lambda model in our affect responsive dialogue

[1] http://help.sentiment140.com/home
[2] https://en.wikipedia.org/wiki/Emoticon

system.

$$
\begin{aligned}
L_{totalSE}(\Theta_e, \Theta_d, \Theta_c, \mathbf{SE}) &= L_{gen1}(\Theta_e, \Theta_d, \mathbf{SE}) + L_{disc1}(\Theta_c) + \boldsymbol{\lambda} L_{disc2}(\Theta_e) \\
L_{totalMD}(\Theta_e, \Theta_d, \Theta_c) &= L_{gen2}(\Theta_e, \Theta_d) + L_{disc1}(\Theta_c) + \boldsymbol{\lambda} L_{disc2}(\Theta_e) \\
&; \boldsymbol{\lambda} \in [0.0, 0.5]
\end{aligned} \tag{4.11}
$$



(a) Style-Embedding with Discriminator Model (b) Multi-Decoder with Discriminator Model

Figure 4.1: Transfer Strength vs Content Preservation results for the two models with adversarial loss where each data label represents the corresponding epoch number.

We provide the transfer strength and content preservation metrics for the model with a lambda over the adversarial component in Figure 4.2 and some sampled reconstruction for the epoch that produces good results in Table 4.2.

We also run the style embedding and multi-decoder model without the adversarial discriminator to check the effect of the discriminator on style transfer. The content preservation and transfer strength results obtained over various epochs of training are visualized in Figure 4.3. We also present some sampled reconstruction results obtained by the model in Table 4.3. The result obtained without the discriminator tends to have higher content

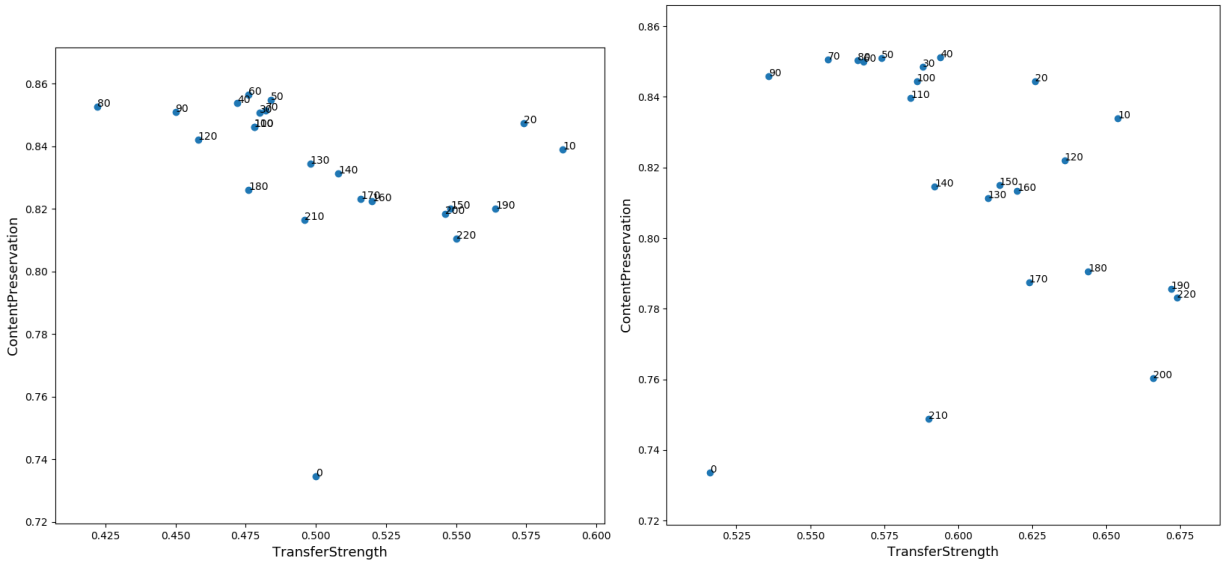
preservation but very poor transfer strength. Hence, we use this model as the baseline in our dialogue system, and it supports our hypothesis that the adversarial discriminator allows for better construction of affect responsive utterance.

4.1a - **Source (Epoch 330)**	**Target Sentiment**	**Reconstruction**
have to write the blog where both great play easy to of mood _EOS_	positive	have to write the blog where both great user myself to shy _EOS_
ugh . i am so sunburned from six flags i had fun , but _EOS_	positive	amazing morning to am so soo chelsea outlet days won had fun awesome fans _EOS_
@ _UNK_ lol u said yr palm was itchy ? thats what it _UNK_ _EOS_	negative	this national _UNK_ post an doctor was in last nasty sorry what hoe can _EOS_
making raw chocolates all morning for the _UNK_ party tonight ! are you coming _EOS_	negative	made the not the morning for the party young - miss out are oy _EOS_
user layin down . i dont feel well _EOS_	positive	sittin be yet . use still think feeling well form well got be _EOS_
4.1b - **Source (Epoch 310)**	**Target Sentiment**	**Reconstruction**
not doing well tonight , saw something on tv , _UNK_ a _UNK_ from _EOS_	positive	hello with your life thanks do twitter msn means here do with bitch _EOS_
tonight was actually kind of fun ! _EOS_	negative	hurting answer ouch god do sorry yet _EOS_
user an op ? ! eep ! i hope she feels better soon ! _EOS_	positive	your wonderful hope everybody now ? hugs have have needs well better ! _EOS_
user i get a chuckle out of the fact that my _UNK_ makes people _EOS_	negative	jb pls what user smile work ? feel work not n't help ? _EOS_
cool ! we 're in june now ! - - - oh ! happy _EOS_	negative	ouch life does u sleep your blackberry ? work ? - ouch _EOS_

Table 4.1: Sample reconstruction results of the best epoch for the Style-Embedding (4.1a) and Multi-Decoder (4.1b) models with adversarial component.

4.2a - **Source (Epoch 10)**	**Target Sentiment**	**Reconstruction**
totally _UNK_ ! we just cnt stop _UNK_ it _EOS_	positive	user safe ! i totally cannot goo _UNK_ it _EOS_
too much pizza , my _UNK_ pants ca n't handle it i think i _EOS_	positive	too much pizza , my _UNK_ _UNK_ are n't even it i think ca _EOS_
4.2a - **Source (Epoch 20)**	**Target Sentiment**	**Reconstruction**
not doing well tonight , saw something on tv , _UNK_ a _UNK_ from _EOS_	positive	is doing as well , it 's the _UNK_ , _UNK_ a _UNK_ from _EOS_
finished ! _EOS_	negative	left tears _EOS_
4.2b - **Source (Epoch 10)**	**Target Sentiment**	**Reconstruction**
user oh dear hope it clears up for you . _EOS_	positive	user lol . thanks for for for you you ._EOS_
user i hugged them all for you ! no movie for now , just _EOS_	negative	user by them already for all you ! no movie for now , i'm _EOS_
4.2b - **Source (Epoch 220)**	**Target Sentiment**	**Reconstruction**
user can i take you out next time so you can judge my future _EOS_	negative	user dont never i ran workout time there so i do numbers follow _EOS_
needs to know what to do when you have water in your ear ! _EOS_	positive	user no know know figure people if do if life in sun ! _EOS_

Table 4.2: Sample reconstruction results of the best epoch for the Style-Embedding (4.2a) and Multi-Decoder (4.2b) models with a lambda to govern adversarial loss.



(a) Style-Embedding with Lambda - Discriminator Model

(b) Multi-Decoder with Lambda - Discriminator Model

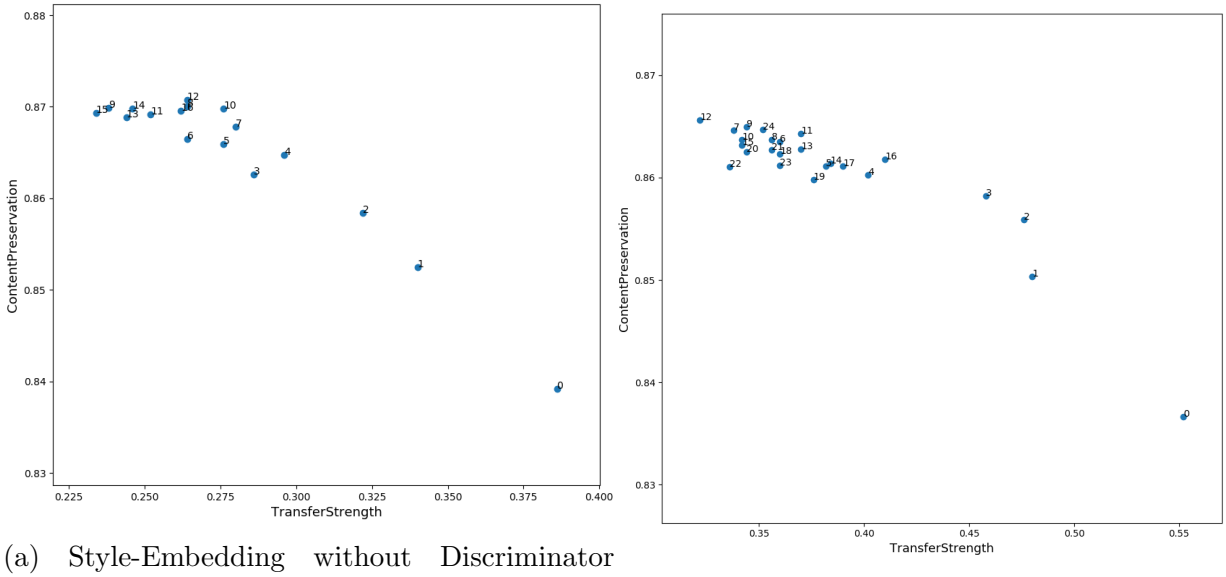
Figure 4.2: Transfer Strength vs Content Preservation results for the two models with a lambda over the adversarial loss where each data label represents the corresponding epoch number.

4.3 Classifier Accuracy

We utilize the classifier model provided by Colneriĉ et al. [17] which is trained on a large collection of Twitter dataset and is used to predict the emotion labels for the Ekman emotion model (Section 2.3.2). The classifier prediction is used to annotate our dialogue dataset with emotion and sentiment labels (Section 3.2). We use this section to evaluate the classifier on two Twitter datasets annotated with emotion and sentiment labels respectively.

For the sentiment annotated Twitter dataset we use the sentiment140 dataset described in the previous section. The sentiment140[3] tweets were annotated using a distant supervision approach by mapping emoticons to their corresponding sentiment label. A random sample of 21,000 Tweets were selected from the sentiment140 dataset to evaluate the classifiers performance on the dataset. We use the NRCTEC tweet dataset [53] (Section 2.3.2)

[3] http://help.sentiment140.com/home



(a) Style-Embedding without Discriminator Model

(b) Multi-Decoder without Discriminator Model

Figure 4.3: Transfer Strength vs Content Preservation results for the two models without the adversarial discriminator where each data label represents the corresponding epoch number.

4.3a - **Source (Epoch 4)**	**Target Sentiment**	**Reconstruction**
watching this bat fly around on the concrete in the earth _UNK_ parking lot_EOS_	positive	watching the day riding on bring in the the the the a _UNK_ total _EOS_
i wanted to win a dsi i tried so hard _UNK_ . i am _EOS_	positive	i wanted to win the mention i accidentally so hard _UNK_ . i am _EOS_
user well , that 's okay . i forgot to send you stuff earlier _EOS_	positive	user well , it 's okay . i didnt to to your messages aswell _EOS_
at the store , trying to budget shop _EOS_	positive	at the , , to to repair store _EOS_
the penguins are falling apart _EOS_	positive	the children are moving falling _EOS_
4.3b - **Source (Epoch 3)**	**Target Sentiment**	**Reconstruction**
user hi sweetie hope you have a good show tonight , good look wish _EOS_	negative	user thank dear hope you have a good night , good you seeing you _EOS_
user i guess the part that looks for location is broken ? _EOS_	positive	user i guess the second works 's stays is is broken ? _EOS_
and no , i'm not crazy , i'm just taking advertising _EOS_	positive	no , , i'm not crazy , i'm just taking yourself _EOS_
user at least you get to watch lost on the way _EOS_	negative	user at least you get to watch the air way to _EOS_
user hiya bec , how are you tonight ? _EOS_	negative	user oh dear , how were you tonight ? _EOS_

Table 4.3: Sample reconstruction results of the best epoch for the Style-Embedding (4.3a) and Multi-Decoder (4.3b) models without the adversarial component.

for testing the accuracy of the classifier on the emotion annotated Twitter dataset. The NRCTEC dataset has a collection of 21,000 Tweets distributed evenly across the six emotion labels in the Ekman emotion model. Table 4.4 shows the prediction accuracy of the classifier on the two datasets.

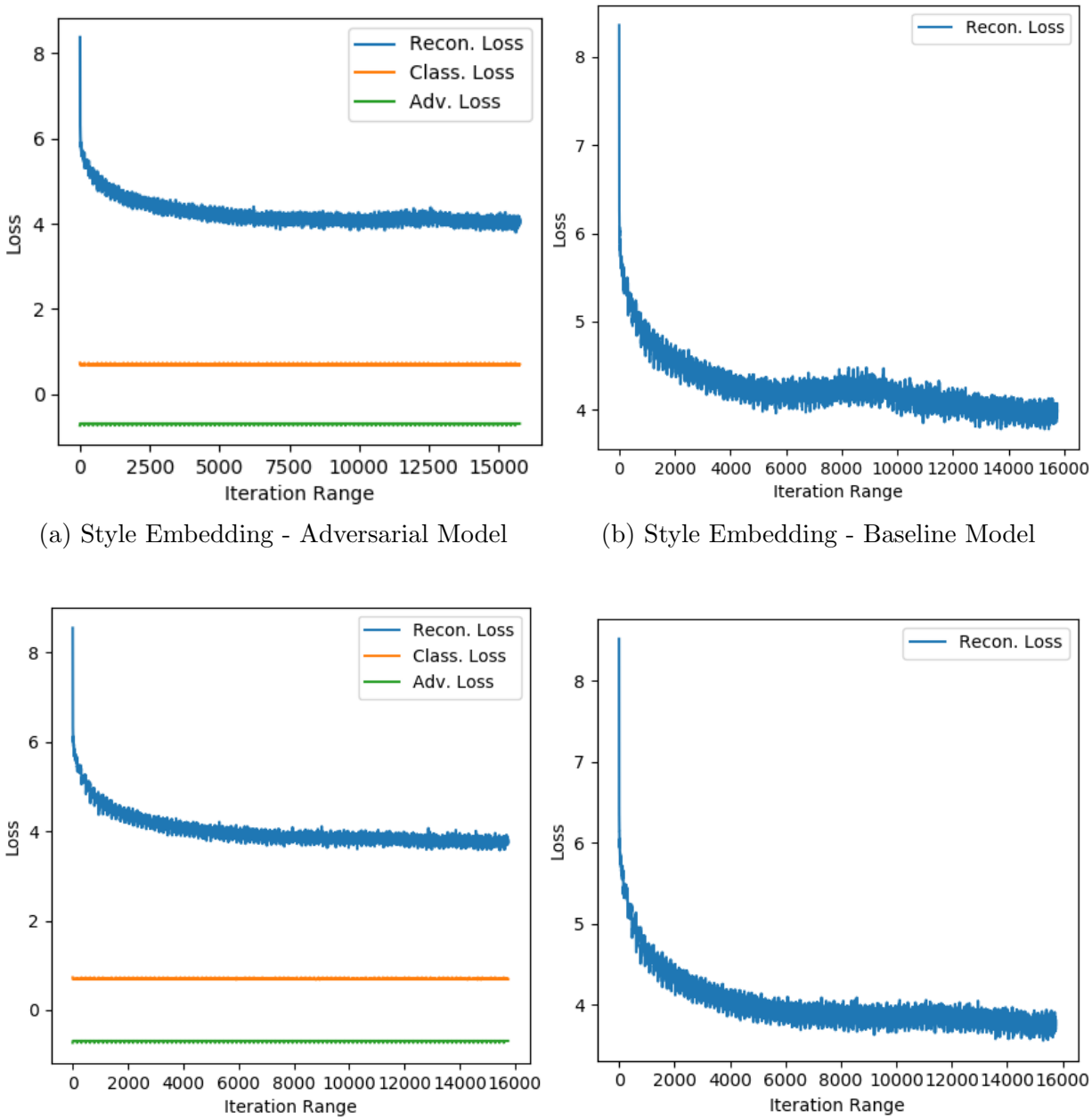
Twitter Dataset	Accuracy
Sentiment140 Dataset	0.6451
NRCTEC Dataset	0.6242

Table 4.4: Classifier accuracy on the Twitter dataset.

4.4 Sentiment Responsive Dialogue System

We present the model component loss obtained during training of our sentiment responsive dialogue system in Figure 4.4. We provide results for two variations of the style embedding and multi-decoder model. The adversarial version of the models is compared with the baseline version which does not contain the adversarial discriminator. The reconstruction, adversarial and classification loss is presented for the adversarial models, while, the reconstruction loss is provided for the baseline models. The reconstruction loss decreases over the training iterations which indicates that the model learns how to generate proper responses.

Figure 4.5 and Figure 4.6 contain the transfer strength vs content preservation and transfer strength vs word overlap scores for all the models, respectively. The number annotation for each data point in the graph denotes the corresponding training epoch number. The transfer strength vs content preservation graph gives us an idea of how well the dialogue model is generating sentiment conditioned responses while maintaining its content information. From the initial assessment of the graph we see that even though the models learn and provide better scores as the training epochs increase, there is a certain level of randomness in the scores for the final epochs. Hence, to select the best epoch for comparison between the models we utilize the model BLEU scores for the generated responses.



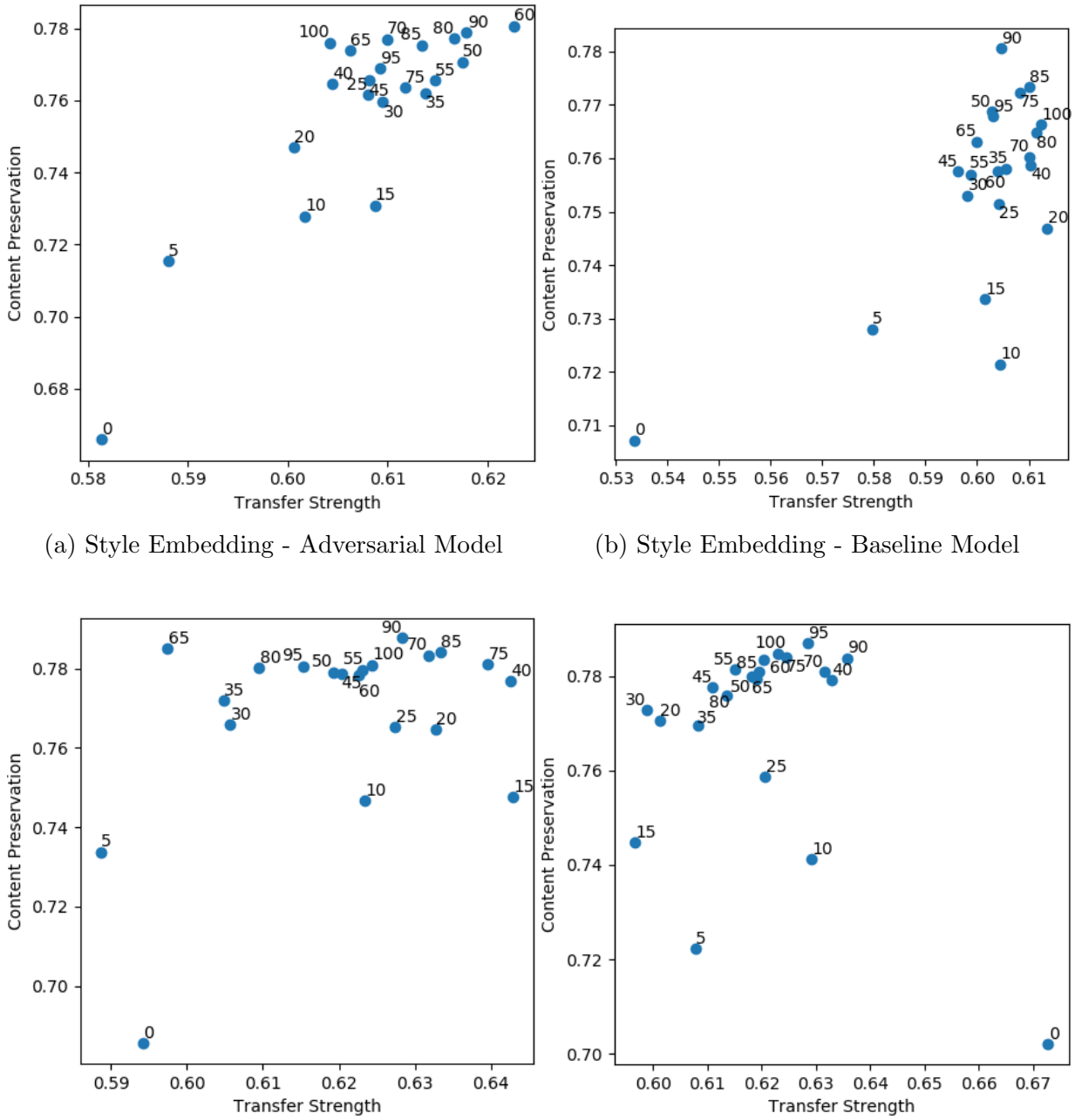
(a) Style Embedding - Adversarial Model

(b) Style Embedding - Baseline Model

(c) Multi Decoder - Adversarial Model

(d) Multi Decoder - Baseline Model

Figure 4.4: Training loss for all sentiment responsive dialogue models.



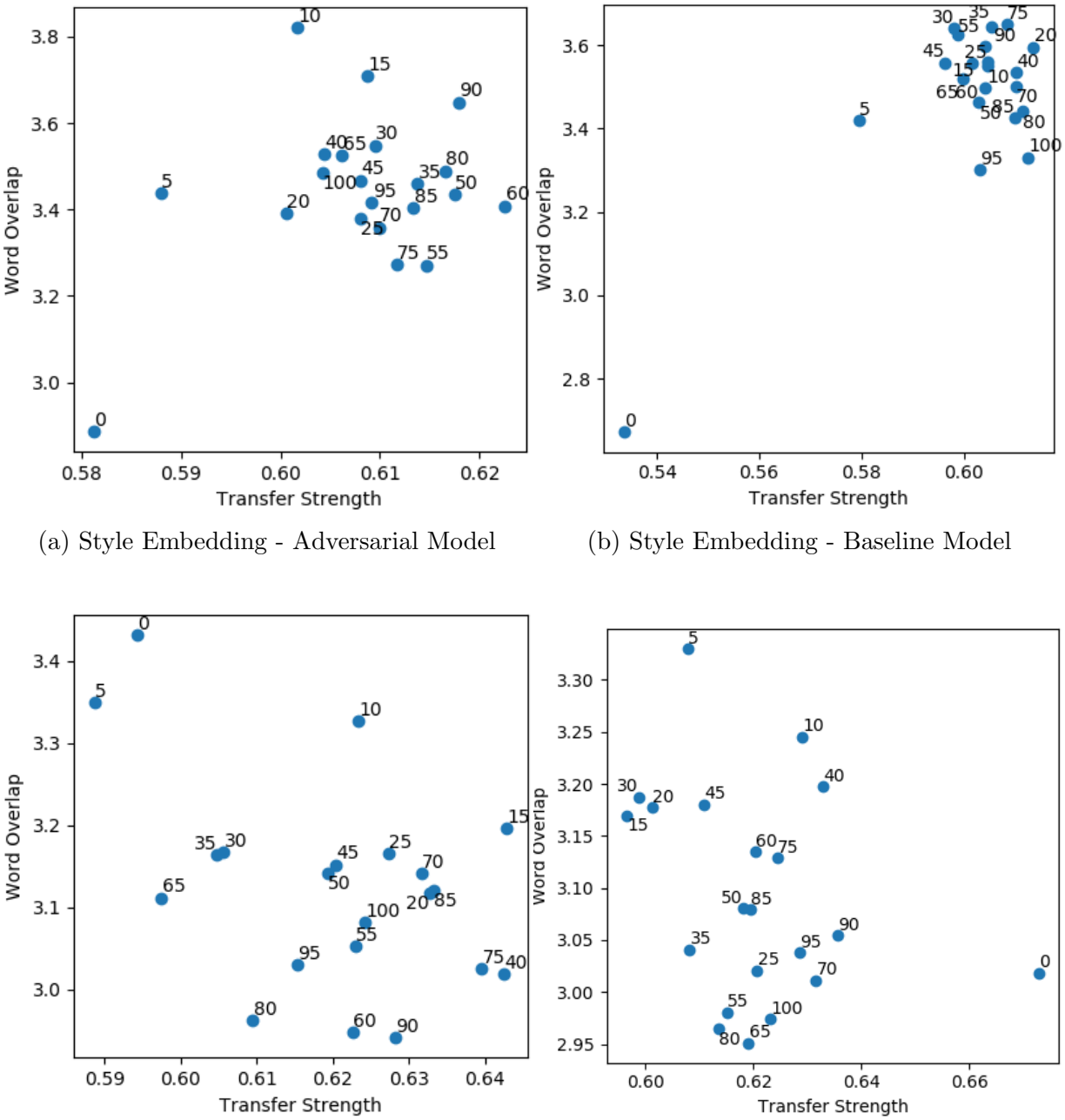
(a) Style Embedding - Adversarial Model

(b) Style Embedding - Baseline Model

(c) Multi Decoder - Adversarial Model

(d) Multi Decoder - Baseline Model

Figure 4.5: Transfer Strength vs Content Preservation results for all sentiment responsive dialogue models where each data label represents the corresponding epoch number.



(a) Style Embedding - Adversarial Model

(b) Style Embedding - Baseline Model

(c) Multi Decoder - Adversarial Model

(d) Multi Decoder - Baseline Model

Figure 4.6: Transfer Strength vs Word Overlap results for all sentiment responsive dialogue models where each data label represents the corresponding epoch number.

We calculate the BLEU score (Section 4.1.4) between the target utterance y_i and the model generated response $y'_{i,j}$ for the actual target affect label s_i (Equation 4.1). The scores are used to select which training epoch of the model provides the best response generation. The BLEU scores for all the sentiment responsive dialogue system task models, obtained over the training epochs is presented in Figure 4.7. The content preservation, transfer strength and word overlap scores for the best training epoch of each model are used to compare all sentiment responsive dialogue models with each other. Table 4.5 shows the training epoch which provides the best utterance generation BLEU scores along with the respective scores.

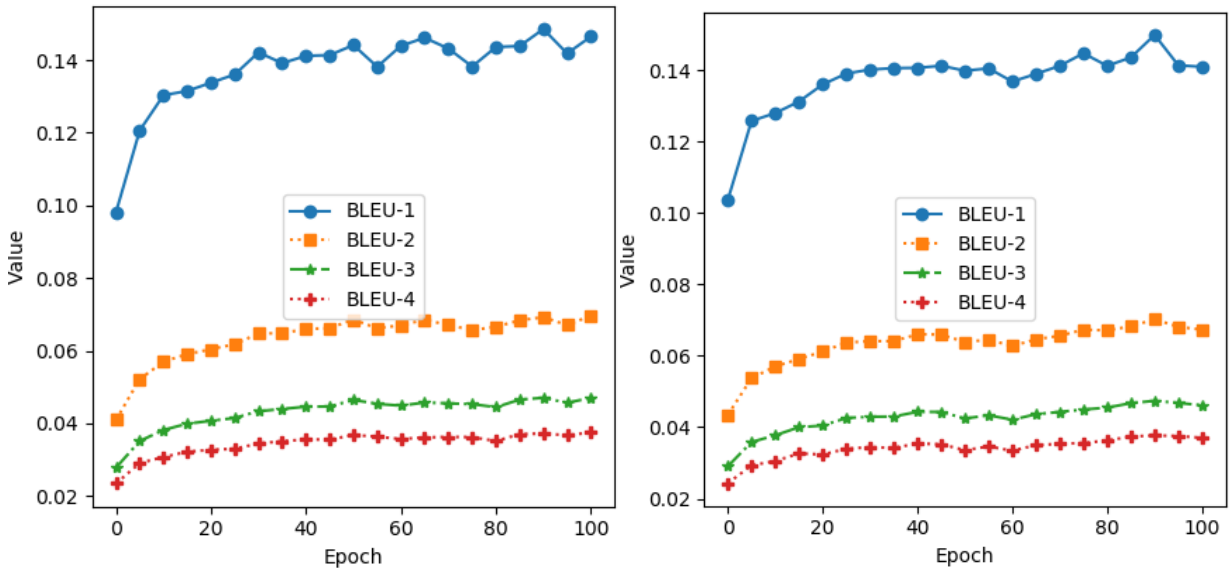
Sentiment Model	Best Epoch	BLEU Score for Best Epoch			
		BLEU-1	BLEU-2	BLEU-3	BLEU-4
Style Embedding - Adversarial Model (SEAdv)	90	0.1485	0.0693	0.0471	0.0372
Style Embedding - Baseline Model (SEBas)	90	0.1499	0.0703	0.0474	0.0377
Multi Decoder - Adversarial Model (MDAdv)	100	0.1550	0.0759	0.0525	0.0418
Multi Decoder - Baseline Model (MDBas)	95	**0.1551**	**0.0762**	**0.0529**	**0.0423**

Table 4.5: BLEU scores of the best epoch for all sentiment responsive dialogue models.

4.5 Emotion Responsive Dialogue System

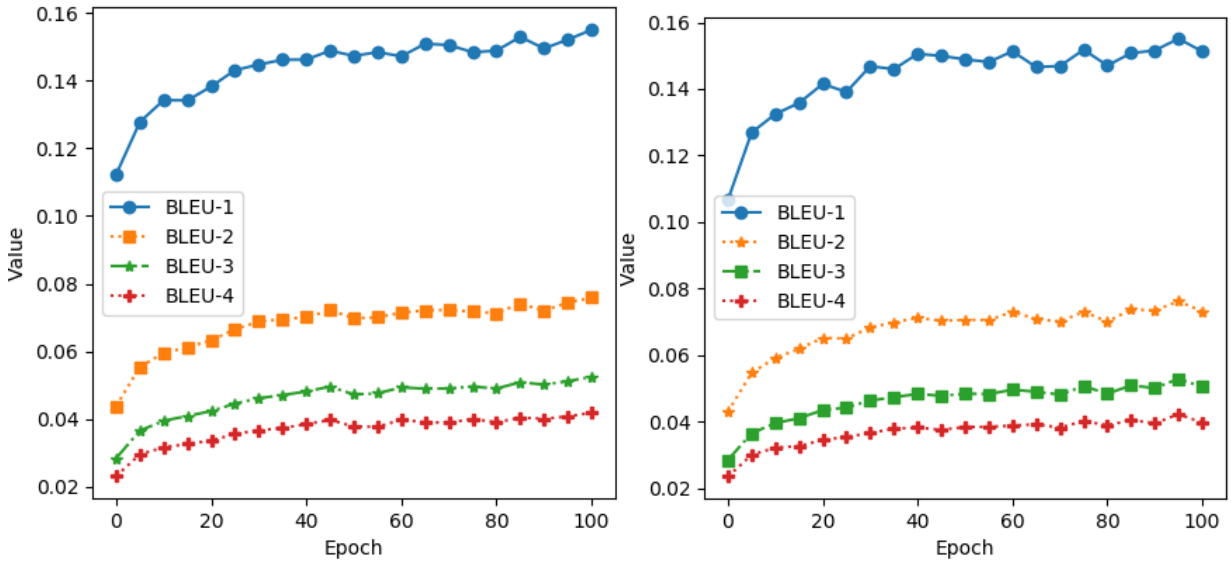
The training loss for all the emotion responsive dialogue systems is presented in Figure 4.8. Similar to the results of the sentiment dialogue system provided in the last section, we provide results for two variations of the style embedding and multi-decoder model used for the emotion responsive dialogue system.

Figure 4.9 and Figure 4.10 contain the transfer strength vs content preservation and transfer strength vs word overlap scores for all the emotion models, respectively. The number annotation for each data point in the graph denotes the corresponding training epoch number. The scores for the emotion dialogue system models show that the efficiency of the models increase almost linearly with training on the transfer strength vs content preservation metric (Figure 4.9).



(a) Style Embedding - Adversarial Model

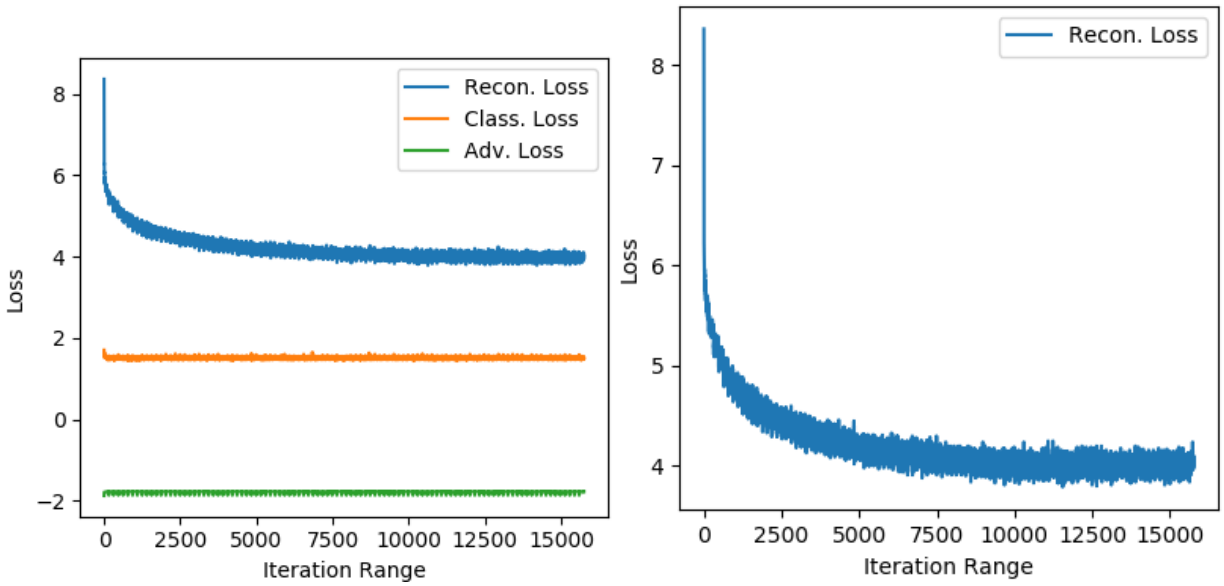
(b) Style Embedding - Baseline Model



(c) Multi Decoder - Adversarial Model

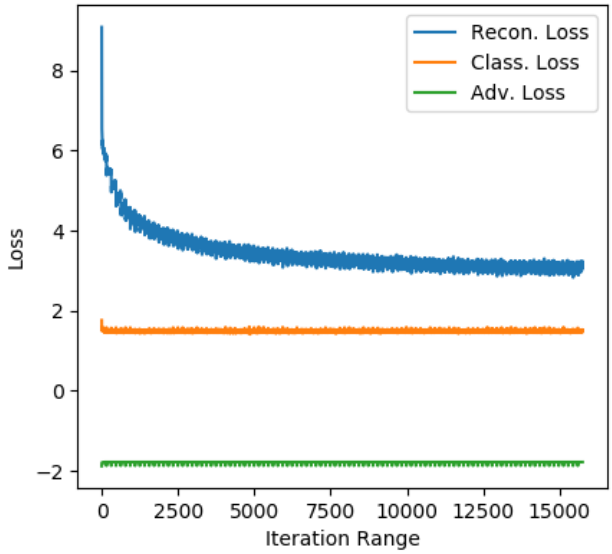
(d) Multi Decoder - Baseline Model

Figure 4.7: BLEU scores to evaluate which epoch generates the best sentiment responsive dialogue response.

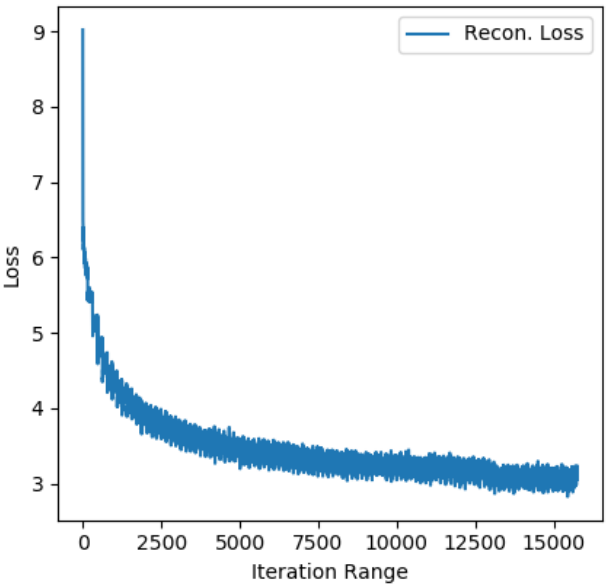


(a) Style Embedding - Adversarial Model

(b) Style Embedding - Baseline Model

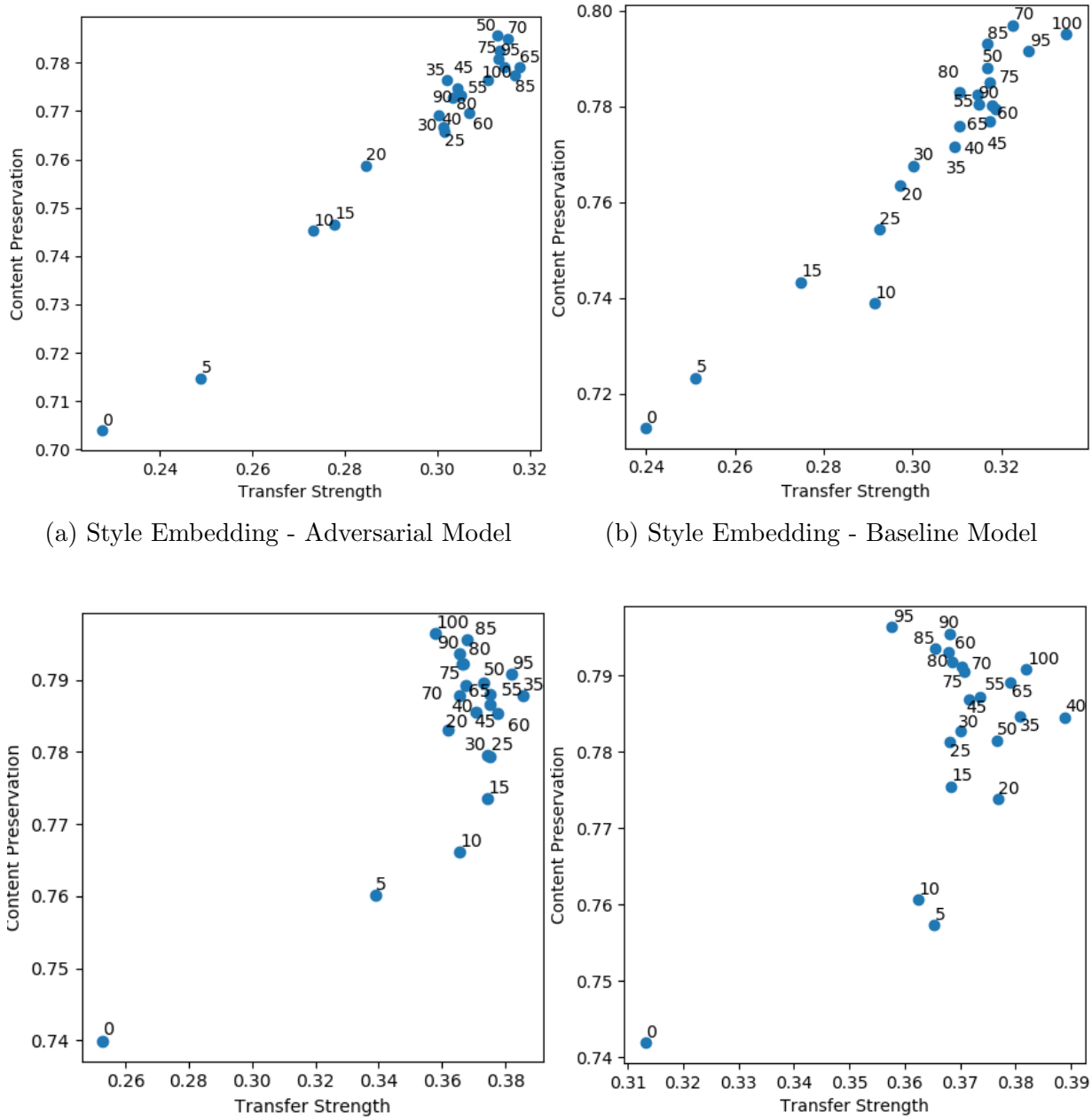


(c) Multi Decoder - Adversarial Model



(d) Multi Decoder - Baseline Model

Figure 4.8: Training loss for all emotion responsive dialogue models.



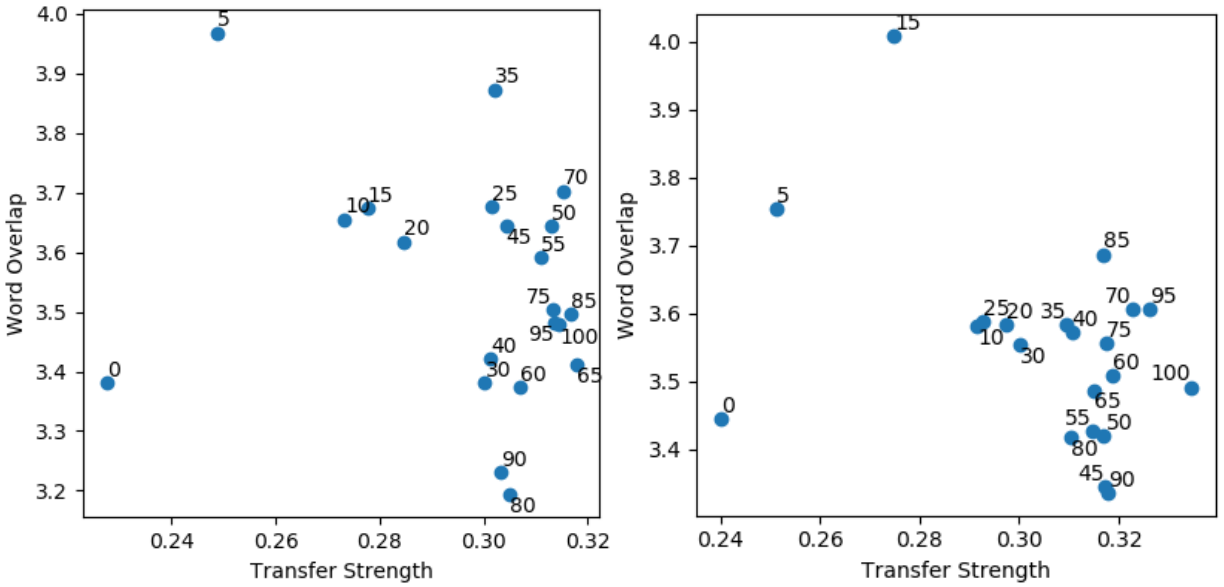
(a) Style Embedding - Adversarial Model

(b) Style Embedding - Baseline Model

(c) Multi Decoder - Adversarial Model

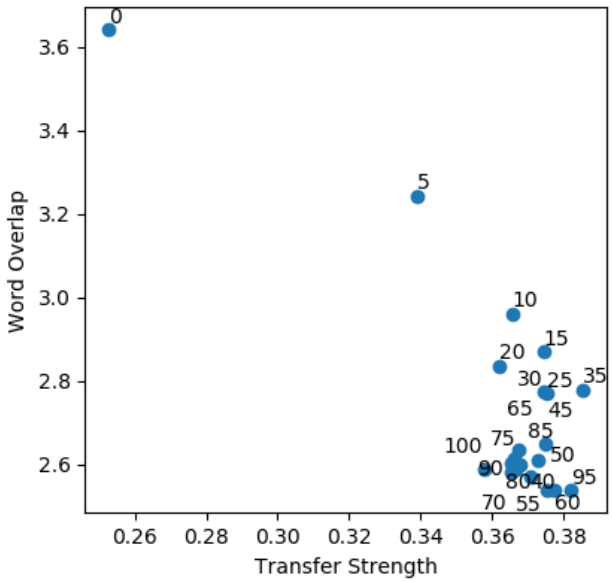
(d) Multi Decoder - Baseline Model

Figure 4.9: Transfer Strength vs Content Preservation results for all emotion responsive dialogue models where each data label represents the corresponding epoch number.

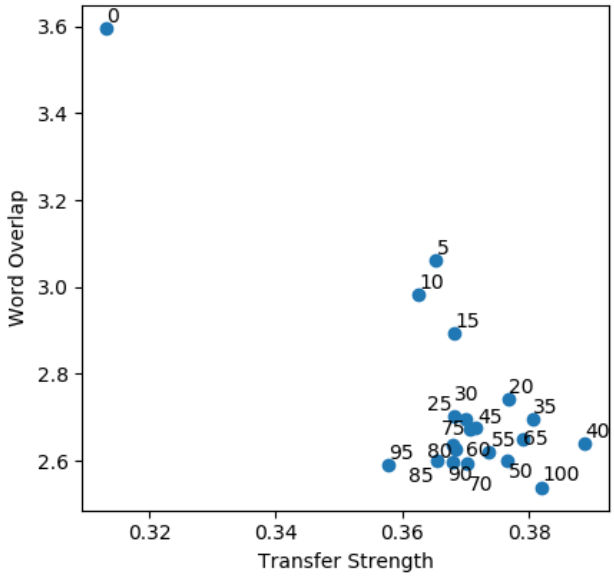


(a) Style Embedding - Adversarial Model

(b) Style Embedding - Baseline Model

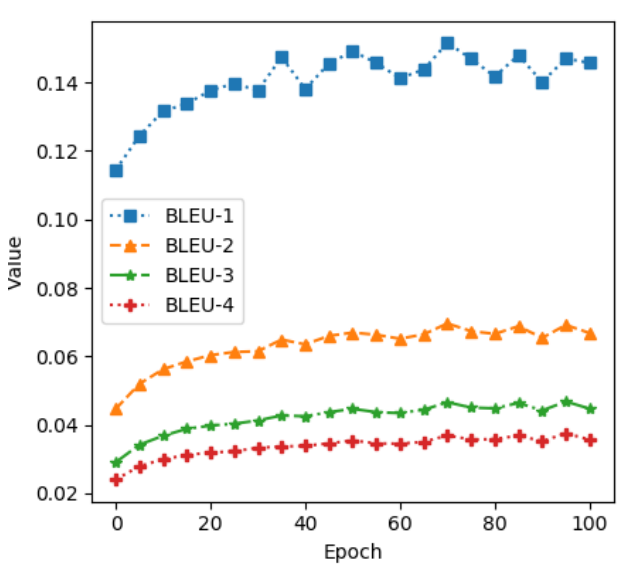


(c) Multi Decoder - Adversarial Model

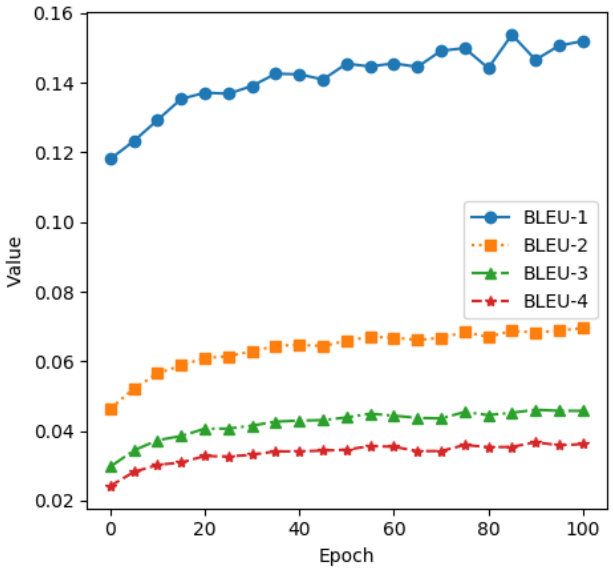


(d) Multi Decoder - Baseline Model

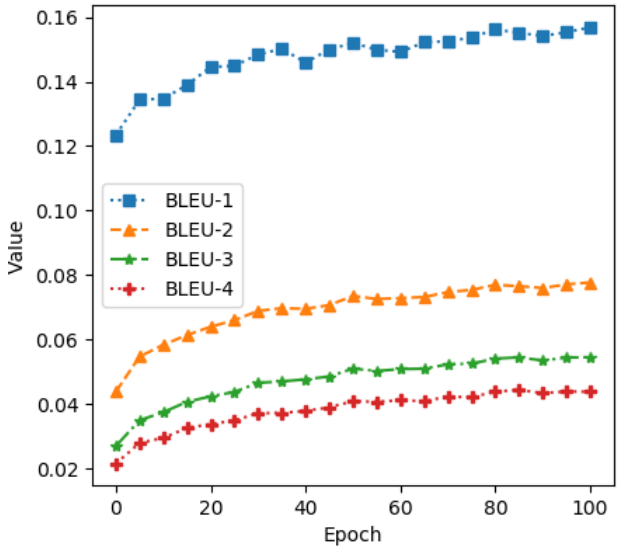
Figure 4.10: Transfer Strength vs Word Overlap results for all emotion responsive dialogue models where each data label represents the corresponding epoch number.



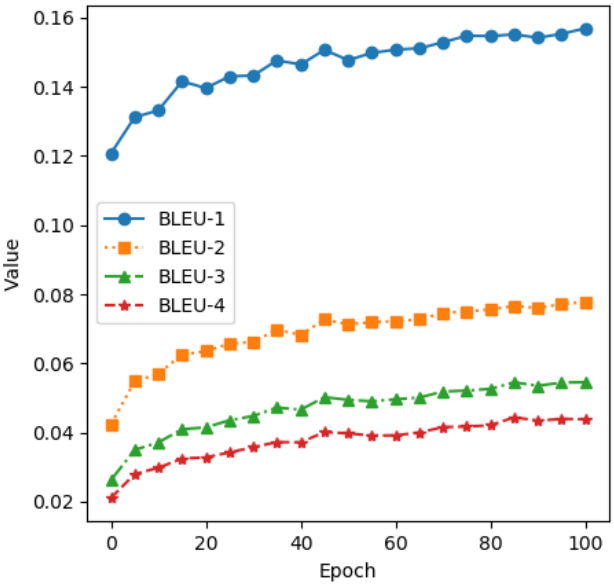
(a) Style Embedding - Adversarial Model



(b) Style Embedding - Baseline Model



(c) Multi Decoder - Adversarial Model



(d) Multi Decoder - Baseline Model

Figure 4.11: BLEU scores to evaluate which epoch generates the best emotion responsive dialogue response.

However, similar to the sentiment dialogue system scores, there is a certain level of randomness in the scores for the final epochs. Hence, we use the model BLEU scores for the generated responses to select the best epoch for comparison between the models . The model response BLEU scores are used to select which training epoch of the model provides the best response generation. The BLEU scores for all the emotion responsive dialogue system task models, obtained over the training epochs is presented in Figure 4.11. The content preservation, transfer strength and word overlap scores for the best training epoch of each model are used to compare all emotion responsive dialogue models with each other. Table 4.6 shows the training epoch which provides the best utterance generation BLEU scores along with the respective scores.

Emotion Model	Best Epoch	BLEU Score for Best Epoch			
		BLEU-1	BLEU-2	BLEU-3	BLEU-4
Style Embedding - Adversarial Model (SEAdv)	95	0.1470	0.0692	0.0468	0.0376
Style Embedding - Baseline Model (SEBas)	90	0.1466	0.0682	0.0461	0.0368
Multi Decoder - Adversarial Model (MDAdv)	100	**0.1568**	**0.0777**	**0.0546**	**0.0439**
Multi Decoder - Baseline Model (MDBas)	100	0.1553	0.0771	0.0544	0.0439

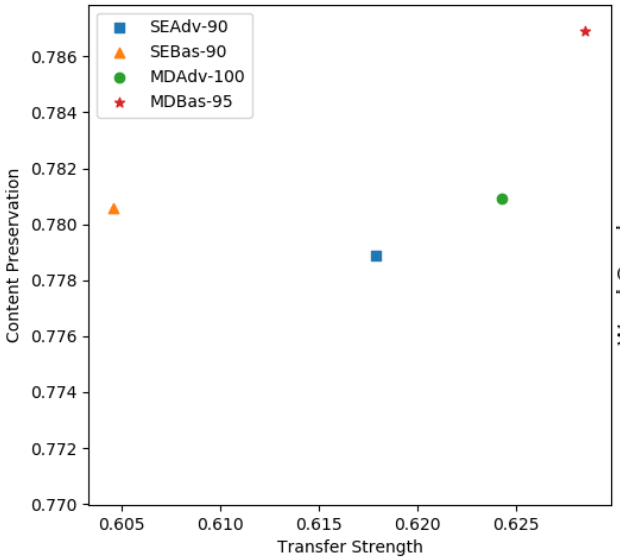
Table 4.6: BLEU scores of the best epoch for all emotion responsive dialogue models.

4.6 Result Analysis

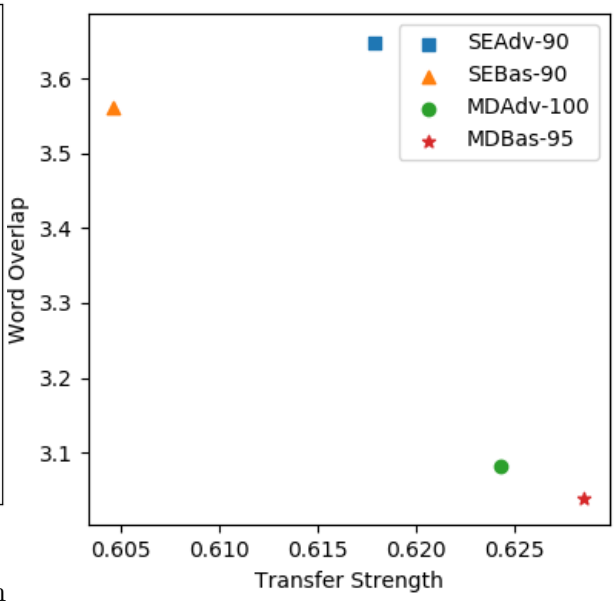
We compare the adversarial and baseline version of the style embedding and multi decoder models using the transfer strength vs content preservation and transfer strength vs word overlap metric (Section 4.1).

Figure 4.12 shows the comparison between all sentiment responsive dialogue models. Even though the difference is minute, we observe that for the sentiment responsive dialogue models the multi decoder model performs better than the style embedding model for content preservation and transfer strength while it achieves lower scores for the word overlap metric. This makes us believe that separating the decoders for each sentiment is able to generate sentiment conditioned responses much better. However, in practice it takes longer

to train the multi decoder model which might make the style embedding model a better choice due to the minute metric score difference.



(a) Transfer Strength vs Content Preservation Score



(b) Transfer Strength vs Word Overlap Score

Figure 4.12: Comparative scores of all sentiment responsive dialogue models for the epoch with best BLEU scores.

We also observe that for sentiment responsive dialogue system task, the style embedding model with the adversarial component transfers the sentiment much better. This makes us believe that making the source utterance representation affect invariant allows for the single decoder to generate responses conditioned on sentiment much better. We provide some sample responses generated by all the sentiment responsive dialogue models in Table 4.7.

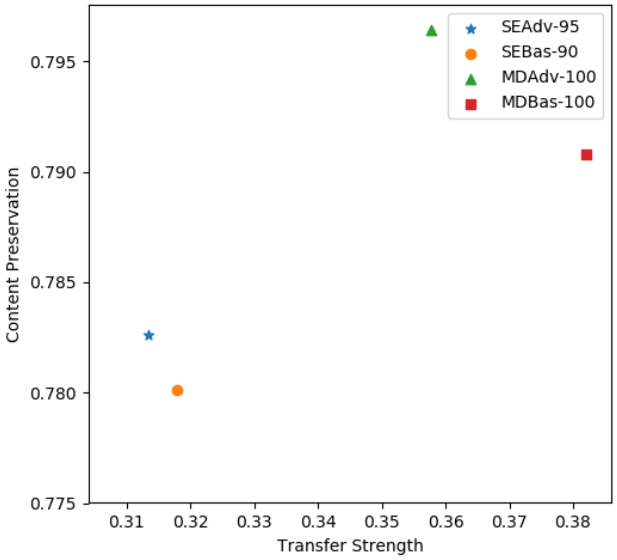
Source Utterance	Target Sentiment	Generated Response
Style Embedding-Adversarial Model, Epoch-90 (**SEAdv-90**) (Figure 4.5a, 4.6a)		
username you have hulu ?	Positive	username i have to pay for
	Negative	username i have no clue
happy birthday to this beautiful soul	Positive	username chan happy birthday you
	Negative	username chan <unk> you are you
Style Embedding-Baseline Model, Epoch-90 (**SEBas-90**) (Figure 4.5b, 4.6b)		
thanks luv , miss you !	Positive	username thank you for sharing !
	Negative	username miss you too much
thoughts so far ? hashtag	Positive	username i deliberately confusing .
	Negative	username i deliberately deliberately trying to get punched in the face .
Multi Decoder-Adversarial Model, Epoch-100 (**MDAdv-100**) (Figure 4.5c, 4.6c)		
happy birthday to this beautiful soul	Positive	username thank you so much !
	Negative	username i love you girly
thanks luv , miss you !	Positive	username thank you miss you
	Negative	username miss you too !
Multi Decoder-Baseline Model, Epoch-95 (**MDBas-95**) (Figure 4.5d, 4.6d)		
username we love you trump	Positive	username trump loves trump !
	Negative	username trump admitted he was a puppet .
username ah ok , ann	Positive	username splendor exactly what i got
	Negative	username you know what you talking about ?

Table 4.7: Sample reconstruction results of the best epoch for all the sentiment responsive dialogue models (Figure 4.12).

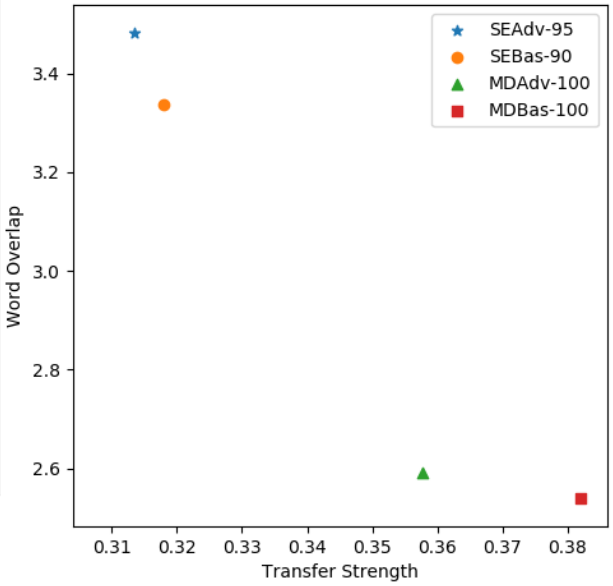
Figure 4.13 shows the comparison between all the emotion responsive dialogue models. From the comparison results, we observe that the multi decoder model performs much better than the style embedding model on the content preservation and transfer strength metric. Thus, similar to the sentiment responsive dialogue system task; separating the decoder for each individual emotion is able to generate emotion conditioned responses much better.

For the emotion responsive style embedding dialogue model, even though the difference is minute, the adversarial component provides a better content preservation score. In practice, the multi decoder model takes a substantially longer time to train than the style embedding model for the emotion responsive dialogue system task. We present some sample responses for all the emotion models in Table 4.8.

Comparing the scores for the sentiment and emotion models we observe that there exists an inverse relation between the content preservation and the word overlap metrics (Figure 4.12, 4.13). An explanation for this might be the fact that a model which is not able to generate diverse responses efficiently has lower content preservation due to responses being short and often containing recurring words. The same response might show better word overlap scores due to the presence of certain similar token words like "username", "hashtag", "<unk>" etc. For the word overlap metric we do not take into account the frequency



(a) Transfer Strength vs Content Preservation Score



(b) Transfer Strength vs Word Overlap Score

Figure 4.13: Comparative scores of all emotion responsive dialogue models for the epoch with best BLEU scores.

of common words which gives the same result for complete responses as well as responses containing recurring words. Upon examination we observe that models with higher word overlap score have poorer transfer strength which validates our assumption. Hence, for a dialogue system architecture we believe that using word overlap metric individually might not be the best evaluation criteria. The adversarial version of both models provide a better word overlap metric score. Unfortunately, due to the minute difference between the metric scores we were not able to see any distinct difference between the adversarial and the baseline version of the models for the affect responsive dialogue system task.

Source Utterance	Target Emotion	Generated Response
Style Embedding-Adversarial Model, Epoch-95 (**SEAdv-95**) (Figure 4.9a, 4.10a)		
loneliness does not suit me	Anger	username i m you
	Disgust	username i don t want to be healthy
	Fear	username i m happiness to you
	Joy	username happiness is happiness life
	Sadness	username i m you loneliness
	Surprise	username i m you
happy birthday to username !	Anger	username aye i was the to
	Disgust	username aye aye aye
	Fear	username <unk> username i love it
	Joy	username aye , happy birthday
	Sadness	username aye i wish i was to
	Surprise	username thank you so much
Style Embedding-Baseline Model, Epoch-90 (**SEBas-90**) (Figure 4.9b, 4.10b)		
username happy new year <unk> !	Anger	username you are to aggressive !
	Disgust	username thanks ! you too !
	Fear	username i hope you too !
	Joy	username happy new year !
	Sadness	username i miss you bunches too
	Surprise	username happy new year !
username i love it !	Anger	username i hate you . i hate it .
	Disgust	username i hate you . i hate it .
	Fear	username i it susie .
	Joy	username i love susie !
	Sadness	username i sealed
	Surprise	username i it susie !
Multi Decoder-Adversarial Model, Epoch-100 (**MDAdv-100**) (Figure 4.9c, 4.10c)		
what's for christmas	Anger	username guys guys are you guys ?
	Disgust	username no one of them .
	Fear	username i am not crying .
	Joy	username the best of the best tradition .
	Sadness	username lol . sometimes .
	Surprise	username juice juice brah ?
happy birthday to username !	Anger	username you don't understand , so please stop
	Disgust	[Example omitted due to profanity]
	Fear	username you you ! you can help you
	Joy	aye aye aye aye aye !
	Sadness	username i wish worked for lambs .
	Surprise	username happy belated anniversary !
Multi Decoder-Baseline Model, Epoch-100 (**MDBas-100**) (Figure 4.9d, 4.10d)		
<unk> nuts is dropping tonight	Anger	username <unk> you re just plain .
	Disgust	[Example omitted due to profanity]
	Fear	username i it s scary
	Joy	username <unk> s a flow
	Sadness	username i m just tweeting
	Surprise	username <unk> you re saving it
i remember that shirt !	Anger	username i hate your kid bout
	Disgust	username i never cheated the tweet
	Fear	username i i get this guy
	Joy	[Example omitted due to profanity]
	Sadness	username i i found out
	Surprise	username <unk> turtleneck shirt .

Table 4.8: Sample reconstruction results of the best epoch for all the emotion responsive dialogue models (Figure 4.13).

Compared to the auto-encoding task's transfer strength results, which show a substantial difference between the adversarial and baseline model (Figure 4.1, 4.2, 4.3) the Seq2Seq model doesn't transfer the affect that effectively. A primary reason for this might be the diverse response options for each source utterance which may or may not contain affect. We also think that our affect annotation step (Section 3.2) might be responsible for poor transfer strength metric scores. We annotate each utterance in our Twitter acquired dialogue dataset with the corresponding affect (emotion and sentiment) label even though some utterance might be void of any affect. Another reason for poor results on the sentiment responsive dialogue system task might be due to the dataset being annotated using an emotion classifier. A final reason might be the uneven distribution of the dataset across the different affect labels during training which might not be sufficient to distinguish between the different affect labels. Using better dataset annotation and training dataset selection steps might provide better results for the affect responsive dialogue system task.

Chapter 5

Conclusion and Future Work

5.1 Summary

In this thesis, we present a dialogue system that can generate responses conditioned on different affect (sentiment/emotion) labels. We design an end-to-end trainable Seq2Seq model inspired by previous work on auto-encoding sentiment style transfer task [27]. We utilize an adversarial learning component to train two models on a Twitter gathered and classifier annotated dialogue dataset. The adversarial learning component is used to make the latent representation of the source utterance affect invariant with the assumption that it would help the models generate responses conditioned primarily on the user specified affect labels. We believe that making the source representation affect invariant might omit any response generation bias on the source utterance affect label.

We ran experiments comparing the adversarial model with a baseline model without an adversarial discriminator and compare the results using three metrics, namely, transfer strength, content preservation and word overlap. Even though the difference is minute, we observe that the models augmented with adversarial learning have better word overlap scores. The style embedding model with adversarial learning provides better affect transfer strength for the sentiment dialogue system task while it has better content preservation for the emotion dialogue system task. Comparing the two type of dialogue models we observe that the multi decoder model performs better than the style embedding model for the affect responsive dialogue system task.

5.2 Future Work

Even though the model is able to generate affect conditioned responses to some degree of success, they are far from optimal. Even though there exist multiple directions for future work, we use this section to describe certain approaches that might improve our model as well as certain tasks where this approach could be used.

- Even though we tried various parameters and model components we believe certain modifications might provide better response generation. Adding a beam search component in the decoder as well as testing other attention mechanisms might make the model generate syntactically better responses.
- Even though Twitter is an excellent source for general cross domain dataset, it often contains text in informal and improper format. This makes the training dialogue dataset quite dirty. Training the models on certain datasets which are written in a formal or semi-formal manner, like the movie dialogue dataset, might provide better results.
- We believe that using the automatic comparison metric like transfer strength and content preservation might not be the best way to evaluate an affect responsive dialogue system model. These metrics are susceptible to poor dialogue generated responses. It would be beneficial to evaluate the responses with a powerful language model to ensure that the sentence is syntactically accurate as well.
- In our assessment the emotion classifier was not able to perform very well on the sentiment domain. Hence, a better Twitter sentiment classifier might be able to annotate and classify the tweets better.
- The style transfer model can be modified into a hierarchical model trained on a multi-turn dialogue dataset so that it allows longer and meaningful multi-turn conversations.
- Another useful technique might be to use a variational Seq2Seq model which might allow the model to generate diverse responses.

References

[1] Charu C Aggarwal and ChengXiang Zhai. A survey of text classification algorithms. In *Mining text data*, pages 163–222. Springer, 2012.

[2] Cecilia Ovesdotter Alm and Richard Sproat. Emotional sequencing and development in fairy tales. In *International Conference on Affective Computing and Intelligent Interaction*, pages 668–674. Springer, 2005.

[3] Saima Aman and Stan Szpakowicz. Identifying expressions of emotion in text. In *International Conference on Text, Speech and Dialogue*, pages 196–205. Springer, 2007.

[4] Nabiha Asghar, Pascal Poupart, Jesse Hoey, Xin Jiang, and Lili Mou. Affective neural response generation. In *European Conference on Information Retrieval*, pages 154–166. Springer, 2018.

[5] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[6] Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. Scheduled sampling for sequence prediction with recurrent neural networks. In *Advances in Neural Information Processing Systems*, pages 1171–1179, 2015.

[7] Daniel G Bobrow, Ronald M Kaplan, Martin Kay, Donald A Norman, Henry Thompson, and Terry Winograd. Gus, a frame-driven dialog system. *Artificial intelligence*, 8(2):155–173, 1977.

[8] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606*, 2016.

[9] Samuel R Bowman, Luke Vilnis, Oriol Vinyals, Andrew M Dai, Rafal Jozefowicz, and Samy Bengio. Generating sentences from a continuous space. *arXiv preprint arXiv:1511.06349*, 2015.

[10] Keith Carlson, Allen Riddell, and Daniel Rockmore. Zero-shot style transfer in text using recurrent neural networks. *arXiv preprint arXiv:1711.04731*, 2017.

[11] Xinchi Chen, Zhan Shi, Xipeng Qiu, and Xuanjing Huang. Adversarial multi-criteria learning for chinese word segmentation. *arXiv preprint arXiv:1704.07556*, 2017.

[12] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[13] Jacob Cohen. A coefficient of agreement for nominal scales. *Educational and psychological measurement*, 20(1):37–46, 1960.

[14] Kenneth Mark Colby. *Artificial paranoia: A computer simulation of paranoid processes*, volume 49. Elsevier, 2013.

[15] Kenneth Mark Colby, Franklin Dennis Hilf, Sylvia Weber, and Helena C Kraemer. Turing-like indistinguishability tests for the validation of a computer simulation of paranoid processes. *Artificial Intelligence*, 3:199–221, 1972.

[16] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12(Aug):2493–2537, 2011.

[17] Niko Colnerič and Janez Demsar. Emotion recognition on twitter: Comparative study and training a unison model. *IEEE Transactions on Affective Computing*, 2018.

[18] Roddy Cowie and Randolph R. Cornelius. Describing the emotional states that are expressed in speech. *Speech Communication*, 40(1-2):5–32, 2003.

[19] Rahul Dey and Fathi M Salemt. Gate-variants of gated recurrent unit (gru) neural networks. In *Circuits and Systems (MWSCAS), 2017 IEEE 60th International Midwest Symposium on*, pages 1597–1600. IEEE, 2017.

[20] R. J. Dolan. Emotion, cognition, and behavior. *Science*, 298(5596):1191–1194, November 2002.

[21] Paul Ekman. Biological and cultural contributions to body and facial movement. pages 34–84, 1977.

[22] Paul Ekman. Are there basic emotions? 1992.

[23] Ekman,P. *An Argument for Basic Emotions.*, pages 169–200. Lawrence Erlbaum, 1992.

[24] Dumitru Erhan, Yoshua Bengio, Aaron Courville, Pierre A. Manzagol, Pascal Vincent, and Samy Bengio. Why Does Unsupervised Pre-training Help Deep Learning? *J. Mach. Learn. Res.*, 11:625–660, March 2010.

[25] Bjarke Felbo, Alan Mislove, Anders Sgaard, Iyad Rahwan, and Sune Lehmann. Using millions of emoji occurrences to learn any-domain representations for detecting sentiment, emotion and sarcasm. *CoRR*, abs/1708.00524, 2017.

[26] J. R. Firth. A synopsis of linguistic theory 1930-55. 1952-59:1–32, 1957.

[27] Zhenxin Fu, Xiaoye Tan, Nanyun Peng, Dongyan Zhao, and Rui Yan. Style transfer in text: Exploration and evaluation. *arXiv preprint arXiv:1711.06861*, 2017.

[28] Albert Gatt and Emiel Krahmer. Survey of the state of the art in natural language generation: Core tasks, applications and evaluation. *Journal of Artificial Intelligence Research*, 61:65–170, 2018.

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

[30] Zellig S Harris. Distributional structure. *Word*, 10(2-3):146–162, 1954.

[31] Maryam Hasan, Emmanuel Agu, and Elke Rundensteiner. Using hashtags as labels for supervised learning of emotions in twitter messages.

[32] Jonathan Herzig, Michal Shmueli-Scheuer, Tommy Sandbank, and David Konopnicki. Neural response generation for customer service based on personality traits. In *Proceedings of the 10th International Conference on Natural Language Generation*, pages 252–256, 2017.

[33] Geoffrey E Hinton, Peter Dayan, Brendan J Frey, and Radford M Neal. The" wake-sleep" algorithm for unsupervised neural networks. *Science*, 268(5214):1158–1161, 1995.

[34] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[35] Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P Xing. Toward controlled generation of text. *arXiv preprint arXiv:1703.00955*, 2017.

[36] Sina Jafarpour and Christopher JC Burges. Filter, rank, and transfer the knowledge: Learning to chat. 2010.

[37] Harsh Jhamtani, Varun Gangal, Eduard Hovy, and Eric Nyberg. Shakespearizing modern language using copy-enriched sequence-to-sequence models. *arXiv preprint arXiv:1707.01161*, 2017.

[38] Dan Jurafsky and James H Martin. *Speech and language processing*, volume 3. Pearson London, 2014.

[39] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[40] Diederik P Kingma, Shakir Mohamed, Danilo Jimenez Rezende, and Max Welling. Semi-supervised learning with deep generative models. In *Advances in Neural Information Processing Systems*, pages 3581–3589, 2014.

[41] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[42] Guillaume Lample, Neil Zeghidour, Nicolas Usunier, Antoine Bordes, Ludovic Denoyer, et al. Fader networks: Manipulating images by sliding attributes. In *Advances in Neural Information Processing Systems*, pages 5967–5976, 2017.

[43] Rémi Lebret, David Grangier, and Michael Auli. Neural text generation from structured data with application to the biography domain. *arXiv preprint arXiv:1603.07771*, 2016.

[44] Anton Leuski and David Traum. Npceditor: Creating virtual human dialogue using information retrieval techniques. *Ai Magazine*, 32(2):42–56, 2011.

[45] Jiwei Li, Michel Galley, Chris Brockett, Georgios P. Spithourakis, Jianfeng Gao, and Bill Dolan. A persona-based neural conversation model. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 994–1003, 2016.

[46] Jiwei Li, Will Monroe, Tianlin Shi, Alan Ritter, and Dan Jurafsky. Adversarial learning for neural dialogue generation. *CoRR*, abs/1701.06547, 2017.

[47] Juncen Li, Robin Jia, He He, and Percy Liang. Delete, retrieve, generate: A simple approach to sentiment and style transfer. *arXiv preprint arXiv:1804.06437*, 2018.

[48] Minh-Thang Luong, Hieu Pham, and Christopher D Manning. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*, 2015.

[49] Bilyana Martinovski and David Traum. Breakdown in human-machine interaction: the error is the clue. In *Proceedings of the ISCA tutorial and research workshop on Error handling in dialogue systems*, pages 11–16, 2003.

[50] Douglas M McNair, Leo F Droppleman, and Maurice Lorr. *Edits manual for the profile of mood states: POMS*. Edits, 1992.

[51] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[52] Kaixiang Mo, Shuangyin Li, Yu Zhang, Jiajun Li, and Qiang Yang. Personalizing a dialogue system with transfer reinforcement learning. *arXiv preprint arXiv:1610.02891*, 2016.

[53] Saif M Mohammad. # emotional tweets. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics-Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation*, pages 246–255. Association for Computational Linguistics, 2012.

[54] Saif M Mohammad and Peter D Turney. Crowdsourcing a word–emotion association lexicon. *Computational Intelligence*, 29(3):436–465, 2013.

[55] Myriam D Munezero, Calkin Suero Montero, Erkki Sutinen, and John Pajunen. Are they different? affect, feeling, emotion, sentiment, and opinion detection in text. *IEEE transactions on affective computing*, 5(2):101–111, 2014.

[56] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 807–814, 2010.

[57] Tong Niu and Mohit Bansal. Polite dialogue generation without parallel data. *arXiv preprint arXiv:1805.03162*, 2018.

[58] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics, 2002.

[59] Nanyun Peng, Marjan Ghazvininejad, Jonathan May, and Kevin Knight. Towards controllable story generation. In *Proceedings of the First Workshop on Storytelling*, pages 43–49, 2018.

[60] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

[61] Robert Plutchik. A general psychoevolutionary theory of emotion. *Theories of emotion*, 1:3–31, 1980.

[62] Shrimai Prabhumoye, Yulia Tsvetkov, Ruslan Salakhutdinov, and Alan W Black. Style transfer through back-translation. *arXiv preprint arXiv:1804.09000*, 2018.

[63] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[64] Sai Rajeswar, Sandeep Subramanian, Francis Dutil, Christopher Pal, and Aaron Courville. Adversarial generation of natural language. *arXiv preprint arXiv:1705.10929*, 2017.

[65] Kirk Roberts, Michael A. Roach, Joseph Johnson, Josh Guthrie, and Sanda M. Harabagiu. Empatweet: Annotating and detecting emotions on twitter. In Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Mehmet Ugur Dogan, Bente Maegaard, Joseph Mariani, Jan Odijk, and Stelios Piperidis, editors, *LREC*, pages 3806–3813. European Language Resources Association (ELRA), 2012.

[66] James A Russell and Albert Mehrabian. Evidence for a three-factor theory of emotions. *Journal of research in Personality*, 11(3):273–294, 1977.

[67] Cicero Nogueira dos Santos, Igor Melnyk, and Inkit Padhi. Fighting offensive language on social media with unsupervised text style transfer. *arXiv preprint arXiv:1805.07685*, 2018.

[68] Klaus R Scherer and Harald G Wallbott. Evidence for universality and cultural variation of differential emotion response patterning. *Journal of personality and social psychology*, 66(2):310, 1994.

[69] Jürgen Schmidhuber. Deep learning in neural networks: An overview. *Neural networks*, 61:85–117, 2015.

[70] Iulian Vlad Serban, Ryan Lowe, Peter Henderson, Laurent Charlin, and Joelle Pineau. A survey of available corpora for building data-driven dialogue systems. *arXiv preprint arXiv:1512.05742*, 2015.

[71] Iulian Vlad Serban, Alessandro Sordoni, Yoshua Bengio, Aaron C Courville, and Joelle Pineau. Building end-to-end dialogue systems using generative hierarchical neural network models. In *AAAI*, volume 16, pages 3776–3784, 2016.

[72] Iulian Vlad Serban, Alessandro Sordoni, Ryan Lowe, Laurent Charlin, Joelle Pineau, Aaron C. Courville, and Yoshua Bengio. A hierarchical latent variable encoder-decoder model for generating dialogues. In Satinder P. Singh and Shaul Markovitch, editors, *AAAI*, pages 3295–3301. AAAI Press, 2017.

[73] Phillip Shaver, Judith Schwartz, Donald Kirson, and Cary O'Connor. Emotion knowledge: further exploration of a prototype approach. *Journal of Personality and Social Psychology*, 52(6):1061–1086, 1987.

[74] Tianxiao Shen, Tao Lei, Regina Barzilay, and Tommi Jaakkola. Style transfer from non-parallel text by cross-alignment. In *Advances in Neural Information Processing Systems*, pages 6830–6841, 2017.

[75] Richard Socher, John Bauer, Christopher D Manning, et al. Parsing with compositional vector grammars. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 455–465, 2013.

[76] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642, 2013.

[77] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

[78] Carlo Strapparava and Rada Mihalcea. Semeval-2007 task 14: Affective text. In *Proceedings of the 4th International Workshop on the Semantic Evaluations (SemEval 2007)*, Prague, Czech Republic, June 2007.

[79] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, 2014.

[80] Joseph Turian, Lev Ratinov, and Yoshua Bengio. Word representations: a simple and general method for semi-supervised learning. In *Proceedings of the 48th annual meeting of the association for computational linguistics*, pages 384–394. Association for Computational Linguistics, 2010.

[81] Alan M Turing. Computing machinery and intelligence. In *Parsing the Turing Test*, pages 23–65. Springer, 2009.

[82] Ankit Vadehra, Maura R Grossman, and Gordon V Cormack. Impact of feature selection on micro-text classification. *arXiv preprint arXiv:1708.08123*, 2017.

[83] Oriol Vinyals and Quoc V. Le. A neural conversational model. *CoRR*, abs/1506.05869, 2015.

[84] Ke Wang and Xiaojun Wan. Sentigan: Generating sentimental texts via mixture adversarial networks. In *IJCAI*, pages 4446–4452, 2018.

[85] Wenbo Wang, Lu Chen, Krishnaprasad Thirunarayan, and Amit P. Sheth. Harnessing twitter "big data" for automatic emotion identification. In *SocialCom/PASSAT*, pages 587–592. IEEE, 2012.

[86] Bolin Wei, Shuai Lu, Lili Mou, Hao Zhou, Pascal Poupart, Ge Li, and Zhi Jin. Why do neural dialog systems generate short and meaningless replies? a comparison between dialog and translation. *arXiv preprint arXiv:1712.02250*, 2017.

[87] Joseph Weizenbaum. Elizaa computer program for the study of natural language communication between man and machine. *Communications of the ACM*, 9(1):36–45, 1966.

[88] Chen Xing, Yu Wu, Wei Wu, Yalou Huang, and Ming Zhou. Hierarchical recurrent attention network for response generation. In Sheila A. McIlraith and Kilian Q. Weinberger, editors, *AAAI*. AAAI Press, 2018.

[89] Zichao Yang, Zhiting Hu, Chris Dyer, Eric P Xing, and Taylor Berg-Kirkpatrick. Unsupervised text style transfer using language models as discriminators. *arXiv preprint arXiv:1805.11749*, 2018.

[90] Zichao Yang, Zhiting Hu, Ruslan Salakhutdinov, and Taylor Berg-Kirkpatrick. Improved variational autoencoders for text modeling using dilated convolutions. *arXiv preprint arXiv:1702.08139*, 2017.

[91] Wenpeng Yin, Katharina Kann, Mo Yu, and Hinrich Schütze. Comparative study of cnn and rnn for natural language processing. *arXiv preprint arXiv:1702.01923*, 2017.

[92] Wojciech Zaremba and Ilya Sutskever. Learning to execute. *arXiv preprint arXiv:1410.4615*, 2014.

[93] Saizheng Zhang, Emily Dinan, Jack Urbanek, Arthur Szlam, Douwe Kiela, and Jason Weston. Personalizing dialogue agents: I have a dog, do you have pets too? *arXiv preprint arXiv:1801.07243*, 2018.

[94] Ye Zhang, Nan Ding, and Radu Soricut. Shaped: Shared-private encoder-decoder for text style adaptation. *arXiv preprint arXiv:1804.04093*, 2018.

[95] Hao Zhou, Minlie Huang, Tianyang Zhang, Xiaoyan Zhu, and Bing Liu. Emotional chatting machine: Emotional conversation generation with internal and external memory. *arXiv preprint arXiv:1704.01074*, 2017.