A Dialogue Tutoring Corpus and Agent for Teaching Programming

A Dialogue Tutoring Corpus and Agent for Teaching Programming

Christina Puthenkalam
Supervisor: Prof. Mrinmaya Sachan, Co-supervisor: Faeze Brahman
ETH Zurich

June 10, 2021

Abstract

When it comes to learning, one-to-one tutoring is a very effective method. However, in our current educational system, this is not always feasible, since one teacher teaches a whole class and has little capacity to tutor each student separately. With the rise of artificial intelligence, new possibilities emerge. There are already some approaches to building intelligent systems that act like tutors and converse with students. Unfortunately, there are not many large open datasets containing conversational data. In this paper, we introduce a new dataset: the WHJ dataset containing 6K conversations and 18.5K utterances. WHJ is a dataset based on chat messages that were sent between tutors and students on the learning platform White Hat Jr[1]. On this platform, students learn to code using simple coding projects. Each student has a tutor they can ask questions regarding the project via chat. These chat messages and the corresponding information about the project are the basis of WHJ. In this thesis, we analyse different dialogue models whilst working with WHJ and the CIMA dataset by Stasaski et al. [2020].

1 Introduction

When comparing different learning methods, one-to-one tutoring, where a tutor converses with a student, is a very effective one. Tutors can help students by nudging them in the right direction. Students can clarify questions, get feedback or simply get appreciation upon successful completion of a task. However, this is not always possible, since usually, a teacher is responsible for many students and cannot always provide one-to-one tutoring.

With the growing interest and success in Natural Language Generation, there is also a lot of progress in creating dialogue systems which converse with humans, often referred to as chat bots. These chat bots can be applied in various areas, like smart home, online shopping, tech support, and of course in the field of education.

One of the main difficulties when creating conversational systems is the generation of sequences which sound natural and match the context. This means it should be a response or reaction to the previous utterance, whilst being also in accordance with the information shared between the two parties during the past conversation. This type of consistency is often a challenge encountered in natural generation tasks.

In the case of an educational dialogue system, the system additionally has to be empathetic and follow a teaching strategy, which ideally is tailored to the individual student and their progress, as well as to the subject it teaches.

There are several ways to build conversational systems. In the early days, most were text-based, where the user asks a question, and the system tries to pattern-match it with one in the database and retrieves an answer for it, like ELIZA by Weizenbaum [1966]. This works well in the cases, where the interaction between human and system is very limited to a specific topic, e.g. frequently asked

[1] https://www.whitehatjr.com/

questions (FAQ), factual knowledge or online tech support. But when it comes to more challenging tasks, where the system needs to emulate a human, this approach is not enough. This can be remedied with the use of natural language generation, which, however, requires large datasets to train the models.

One field of education, where there is already a lot of progress is language learning. The goal is to create a system that teaches a foreign language to the student by teaching and checking grammar and vocabulary, or by conversing with them.

EnglishBot by Ruan et al. [2021] is such a chat bot which engages with students in spoken conversations to improve their English. Another large dataset, which focuses on teaching English via writing, is the Teacher-Student Chatroom Corpus (TSCC) by Caines et al. [2020]. This dataset is based on one-to-one English lessons in an online chatroom. However, this means that the conversations in TSCC are authentic, which introduces additional challenges like handling informal language and grammatical mistakes.

CIMA by Stasaski et al. [2020] is similar to the aforementioned datasets, but teaches Italian. In the conversations found in the CIMA dataset, the students try to describe a given image in Italian. The tutors correct the description if necessary, provide feedback and help the student by giving hints. The data was collected using crowd workers who role-played the scenario.

The potential of the tutoring systems is not limited to language teaching. AutoTutor by Graesser et al. [2004] is a tutoring system which teaches physics. However, these systems also need a thorough knowledge of the subject they teach. Sánchez Díaz et al. [2018] propose a methodology to build an intelligent tutor with two components which address the modelling of knowledge and the flow of the conversation separately.

We see that there are already some intelligent tutoring systems, but there not many large open datasets with an educational focus.

In this thesis, we introduce a new educational dataset WHJ, which is based on conversations between tutor and student, discussing a coding project (see Section 2.1). The dataset also includes the project description, the student's submitted code and the solution code.

To check and compare the results for this new dataset, we also run the same models with the CIMA dataset (see Section 2.2). We work with two types of models. Firstly, large models which are pre-trained on another big dataset which then can be fine-tuned to the respective dataset for generation (see Section 3.1). We concretely use BART by Lewis et al. [2020] and GPT-2 by Radford et al. [2019]. And secondly, a sequence-to-sequence (seq2seq) model (see Section 3.2), with an encoder-decoder structure and Long Short-Term Memory (LSTM) cells (Hochreiter and Schmidhuber [1997]). We find that our seq2seq model works better for both datasets and produces better results.

2 Datasets

As mentioned above, two tutor-student conversation datasets are used in this thesis: WHJ and CIMA. The following sections explain how the data was collected and show some statistics of both datasets.

2.1 WHJ

White Hat Jr[2] is an online coding platform for kids which provides one-to-one coaching. On the platform, students learn to code by solving small coding projects, during which students can communicate with their tutors via chat.

These tutor-student dialogues regarding a coding project, which the student is trying to solve, form the basis of the new conversation dataset WHJ. Unlike many other datasets, which require crowdworkers to act out a scenario, the conversations in WHJ are authentic. This has its benefits and drawbacks.

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

	Full Dataset	Final Dataset
Number of Conversations	23744	6029
Number of Utterances	36288	18566

Table 1: Statistics of the dataset. Full Dataset refers to the dataset after combining same-speaker utterances and Final Dataset to the dataset after removing single-sentence conversations. Due to the drastic decrease in the number of conversations in the Final Dataset, it is clear that most of the conversations in the Full Dataset only consist of one utterance.

	Full Dataset	Final Dataset
Min length	1	2
Max length	27	27
Mean length	1.5	3
Median length	1	2

Table 2: Statistics about the lengths of the conversations in the respective datasets. Here we see, that even though there are some longer conversation with up to 27 utterances, most conversations in the Final Dataset are shorter with two or three utterances.

A huge advantage is the easy collection of data. Instead of relying on and paying crowdworkers to explicitly contribute to the dataset, conversations are a byproduct of the teaching platform and have almost no overhead.

Another advantage is the fact that the conversations revolve around concrete coding tasks. This means there is additional context information in the form of the coding project description, the student's code, and the solution code. These may also be included as context for a model.

The disadvantage of the authenticity is the randomness and noise. Like in natural conversations not all messages are task-related and goal-oriented. Some are random and may contain grammar mistakes and out-of-context information. They may also include sensitive information about the dialogue partners. Additionally, since the messages do not stem from a controlled environment with a clear task, they vary vastly in terms of length and content. Some are very unique, which may make learning from the dataset harder.

To prepare the dataset, the chat messages belonging to the same projects were combined into conversations, where the labels *Teacher* and *Student* indicate who the speaker is. All consecutive utterances made by the same speaker (e.g. the teacher sends two messages before the students sends one) are combined to one utterance. This results in the Full Dataset, mentioned in Table 1.

Conversations with just one utterance were filtered out, since they are not really useful for a dialogue system. Furthermore, conversations containing utterances longer than the maximum allowed length (250) were removed as well. This Final Dataset after filtering is used for all experiments in the thesis and will hereafter be referred to as WHJ. It contains 6029 conversations and 18'566 utterances in total, as shown in Table 1. In Table 2, we see that even after filtering single utterance conversations, most conversations are short, consisting of two or three utterances. This again may be problematic, if the system needs to generate longer conversations.

To remedy the issue of names, phone numbers and other sensitive and noisy data in WHJ, the dataset was cleaned by introducing labels to replace those. This was only done for the second model. More details on the labels and the concrete preparation of the data for both models can be found in Section 4.2.

An example conversation found in WHJ:

STUDENT: Mam in this project when it touches a healthy food it gives an achievement sound and when it touches a junk food it gives an alert sound
TEACHER: okay Good ... but can you try to change the color of jojo also
STUDENT: Ok mam. Mam i cannot change the costume but i changed the background design
TEACHER: You dont have to change the costume, you have to change the color of jojo. There is a block to change the color for that. just try to find that

2.2 CIMA

The CIMA dataset[3] (Stasaski et al. [2020]) consists of two separate datasets called the Shape Dataset and the Prepositional Phrase Dataset. In this thesis, only the Prepositional Phrase Dataset was used, since it is bigger, and will hereafter be referred to as CIMA.

CIMA was collected using crowdworkers on Amazon Mechanical Turk[4] who role-played as both the Student and the Tutor. They were given an image, a preposition, a color, and two objects in both English and Italian. They also received the past conversation. Students and Tutors then needed to construct a response and a classification (called action label) of their response. For Students the action labels were: *Guess*, *Clarification Question*, *Affirmation*, *Other*. For Tutors, they were: *Hint*, *Question*, *Correction*, *Confirmation*, *Other*.

CIMA holds 1135 different conversations. Each of the conversations has the same past conversation and several tutor responses (from different crowd workers). By combining the past conversation with each of these responses, there are 3315 input-output pairs that can be used for a model.

3 Models

The first approach to create a dialogue system was to use pre-trained models which are fine-tuned on WHJ and CIMA, respectively. Two models were considered: BART by Lewis et al. [2020] and GPT-2 by Radford et al. [2019] (see Section 3.1).

Since they did not perform very well on the datasets, another simpler model was used as a second approach: a sequence-to-sequence (seq2seq) model with an encoder-decoder architecture using LSTM units (see Section 3.2). This model produced better results.

3.1 Pre-trained models

The following sections show the two pre-trained models that were used.

BART: BART is an autoencoder for pre-training sequence-to-sequence models and was introduced by Lewis et al. [2020]. It uses a transformer-based neural machine translation architecture. BART can be used for several different tasks, e.g. sequence and token classification, machine translation and sequence generation, which is the case here. The data used to pre-train BART consists of 16GB of news, books, stories, etc. This, however, means that a lot of memory is needed. Since the GPU used in the experiment did not have sufficient memory to run the full BART model (`bart-large`[5]), a smaller one was used for the experiments in this thesis: `bart-base`[6].

GPT-2: GPT-2, introduced by Radford et al. [2019], is a transformer-based language model which is used to generate text as a response to a sequence given to the model. It was pre-trained mostly in an unsupervised manner on a very large amount of data and then in a supervised manner on a smaller dataset. In total the pre-training data for GPT-2 consists of 40GB of text, where the type of the text varies widely, to create a more diverse dataset.

Here again the same issues with GPU memory as mentioned above were encountered. Therefore a smaller model was used: gpt-medium [7].

3.2 Seq2seq

Our seq2seq model follows closely the model by Stasaski et al. [2020]. It has an encoder-decoder architecture and uses LSTM units.

[3] https://github.com/kstats/CIMA
[4] https://www.mturk.com/
[5] https://huggingface.co/bert-large-uncased
[6] https://huggingface.co/facebook/bart-base
[7] https://huggingface.co/gpt2-medium

Embeddings: We used different embeddings during the experiments: None (i.e. integer encoding, where all words in the corpus are assigned an integer), GloVe (Pennington et al. [2014]) and Word2vec (Mikolov et al. [2013]). In addition, we tried a hybrid format, where the input was embedded using BERT (Devlin et al. [2018]) and the output was only encoded with integer encoding. We find that GloVe generally performed the best.

Encoder: The encoder is a stacked LSTM with a variable number of layers. After the encoder is given the encoder input (i.e. input sequence $x^E_{1:N}$), it encodes it and passes the final hidden and cell states (h_N and c_N), as well as the outputs at each time step ($Y^E = y^E_{1:N}$) to the decoder.

$$y^E_t, h_t, c_t = LSTM(x^E_t, h_{t-1}, c_{t-1})$$

Decoder The LSTM in the decoder is initialized using the final cell and hidden states of the encoder. The decoder is then given the output of the encoder LSTM (Y^E) and the decoder input (i.e. target sequence $x^D_{1:N}$ with a start token at the beginning).

Global attention: In the decoder, we also use global attention (introduced by Bahdanau et al. [2014]). Attention provides a way for the decoder to focus on the more relevant parts of the encoded input. The attention weights are calculated by concatenating (denoted by `CAT`) the decoder inputs and the last hidden state of the encoder and then by applying a linear transformation, as seen in Equation 1. The weights are then multiplied with the encoder outputs (see Equation 2) to obtain a weighted encoded input. In Equation 3, we see how the decoder input is then combined with the weighted encoder output.

$$W_{Attn} = softmax(W_1 * \texttt{CAT}(x^D_t, h_{t-1})) \tag{1}$$

$$Y^E_{Attn} = W_{Attn} * Y^E \tag{2}$$

$$\tilde{x}^D_t = relu(W_2 * \texttt{CAT}(x^D_t, Y^E_{Attn})) \tag{3}$$

$$\tilde{y}^D_t, h_t, c_t = LSTM(\tilde{x}^D_t, h_{t-1}, c_{t-1})$$

$$y^D_t = log(softmax(\tilde{x}^D_t))$$

Copy mechanism: To generate rare words a simple copy mechanism is used. First, rare words (words occurring less than e.g. 3 times) are replaced with an UNK (unknown) token. Then the model is trained normally using these tokens. During inference, any time an unknown token is generated, the token is replaced by the word in the input sequence with the most attention (given by the global attention mentioned in the previous paragraph).

$$attn\_idx = argmax(W_{Attn})$$

$$y^D_t = \begin{cases} x^E_{attn\_idx} & \text{if } y^D_t = \text{UNK} \\ y^D_t & \text{otherwise} \end{cases}$$

4 Experiments and Results

Both datasets (WHJ and CIMA) mentioned in Section 2 were run on both types of models from Section 3. We had two goals: firstly, we wanted to evaluate the performance of our new dataset WHJ. Secondly, since our seq2seq model is based on the model by Stasaski et al. [2020], we wanted to reproduce their results, which we achieved.

4.1 Setup

All experiments were run on the Leonhard cluster[8]. The jobs were run using one GPU with a memory of 11 GiB. As a metric to evaluate the performance of the model, we used BLEU scores introduced by Papineni et al. [2002].

All the code that was used and the results shown in this section can be found on GitHub[9].

4.2 Data Preparation

To be used in the models, the conversations had to be prepared to form pairs of an input and an output sequence. The data preparation varied for the two model types. After the preparation of the pairs, they were split into 3 sets: Train (0.8), Validation (0.1) and Test (0.1).

Pre-trained models: The input consists of the context in form of the past conversations and the output of the utterance that follows. For a conversation with e.g. 4 utterances, 3 input-output-pairs can be created. Once the context gets longer than the maximum allowed length, we move on to the next conversation. This data preparation results in ∼15'000 input-output-pairs for CIMA and ∼12'000 for WHJ. But it is to note, that there are some redundancies, since all input-output-pairs created from the same conversation have common prefixes.

Seq2seq model: Since our seq2seq model closely follows the model described by Stasaski et al. [2020], the preparation of the data was also adopted. Since Stasaski et al. [2020] saw no particular benefit in including the whole past conversation, they limit the number of utterances in the past conversation to 2. Therefore, this is done for both WHJ and CIMA in this thesis. This results in ∼3300 input-output-pairs for CIMA and ∼9000 for WHJ.

Due to the authenticity of WHJ, it contains a lot of names, telephone numbers and other sensitive and noisy data. To remedy that, labels were introduced to replace those: `<NAME>`, `<NUMBER>` (for telephone numbers), `<URL>` and `<EMAIL>`. The introduction of the labels shows an increase in performance.

In the case of CIMA, additionally, a context consisting of the action labels, the preposition, color and object in Italian and in English is added, followed by `EOC` which denotes the end of the context. This results in an input of the following format:

INPUT: Hint, Correction, e di fronte al, giallo, coniglio, is in front of the, yellow, bunny, `EOC`
Tutor: Well, "bunny" is "coniglio" Student: il gatto e di fronte al coniglio.
OUPUT: So very close! Remember your color?

4.3 Results

4.3.1 BART and GPT-2

The limiting factor for the pre-trained models soon proved to be memory. Since the jobs could only be run on 1 GPU with a memory of 11 GiB, smaller pre-trained model had to be used: i.e. `bart-base` instead of `bart-large` and `gpt2-medium` instead of `gpt2-xl`.

Unfortunately, the results obtained using the pre-trained models are quite poor. The generated utterances are very general and often the same, e.g. "Thank you, ma'am", "Okay, ma'am" or "Ok". This naturally leads to poor BLEU scores, e.g. 0.0008. To check if this is an issue of WHJ, the same models were also run with CIMA, which did not yield good results either. The reason for this might be the small size of WHJ and CIMA (in comparison to the dataset the model was pre-trained on), or the manner of the data preparation, which lead to several input-output-pairs with similar input sequences but different output sequences.

In the case of CIMA, the poor results obtained when training with `bart-base` may be the result of CIMA containing Italian sentences, since `bart-base` is intended for English sentences only. In

[8] https://scicomp.ethz.ch/wiki/Leonhard
[9] https://gitlab.ethz.ch/puthenkc/master_thesis_puthenkc

Models	BLEU Score
Simple Model	
Basic configuration	0.193
Enc 3	0.193
WS 6000	0.185
Enc 3, WS 6000	0.183
Enc 3, WS 6000, Bi-LSTM	0.193
Copy Model (min occ 5)	
Basic configuration	0.193
Enc 3	0.196
WS 6000	0.196
Enc 3, WS 6000	0.192
Enc 3, WS 6000, Bi-LSTM	**0.209**
Embeddings	
None	0.191
Word2vec	0.205

Table 3: Results of the parameter tuning for our seq2seq model on the WHJ dataset. The best score was achieved with the Copy Model with 3 encoding layers, warm start and using bi-directional LSTM cells. If a word occurred less than 5 times in the whole corpus, it was replaced with an unknown token. This model was also run with different embeddings to evaluate their impact on the BLEU score.

this instance, it would actually be better to use mBART (Liu et al. [2020]), which is the multilingual version of BART. However, the used GPUs did not have enough memory to support mBART, which is larger than `bart-base`, due to the multilingualism.

4.3.2 Seq2seq model

Since our seq2seq model closely follows the model used by Stasaski et al. [2020], the goal was to reproduce their results on the CIMA dataset. This was reached with a best BLEU score of 0.328, which is slightly higher than the best score of 0.31 reported by Stasaski et al. [2020].

The BLEU scores obtained for different model settings for WHJ and CIMA can be found in the Tables 3 and 4. The basic configuration (BC) denotes the model, with one encoder and one decoder layer. The hidden size of the LSTM is 256, and the model uses the Adam optimizer (with a learning rate of 0.001). "WS" stands for warm start and the number after WS shows for how many steps a warm start was performed. We also tried early-stopping, but that did not show good results, thus it is not included in the table. Bi-LSTM denotes the use of bi-directional LSTMs. In terms of stacking the LSTM, only adding more layers to the encoder showed good results, with the best number of layers being 3. Copy Model denotes the model where the copy mechanism described in Section 3.2 was used.

Three types of embeddings were used: none at all (only integer-encoding), GloVe and Word2vec. Here GloVe generally performed better than the other two.

WHJ: The seq2seq model works relatively well with WHJ. Here again, in some cases it suffers from the problem, that a very general sequence is generated. This probably has to do with the fact that some utterances are very unique due to the authenticity and thus difficult to learn from and generalize.

The best observed BLEU score was 0.209 (see Table 3), with the Copy Model, using 3 encoding layers, warm start and bi-directional LSTM cells. The use of GloVe embeddings showed the best results.

Models	BLEU Score
Simple Model	
Basic configuration	0.257
Enc 3	0.249
Enc 3, WS 6000	0.312
Enc 3, WS 6000, Bi-LSTM	**0.328**
Copy Model (min occ 3)	
Basic configuration	0.253
Enc 3	0.257
Enc 3, WS 6000	0.310
Enc 3, WS 6000, Bi-LSTM	0.296
Embeddings	
None	0.297
Word2vec	0.289

Table 4: Results of the parameter tuning for our seq2seq model on the CIMA dataset. The best model was achieved using the Simple Model with 3 encoding layers, warm start and bi-directional LSTM cells. This model was also run with different embeddings to evaluate their impact on the BLEU score.

CIMA: The results of the parameter tuning for CIMA can be found in Table 4. The best score of 0.328 (comparable to the score of 0.31 reported by Stasaski et al. [2020]) was obtained using the Simple Model, with 3 encoding layers, warm start and bi-directional LSTM cells. Here again, using GloVe embedding showed the best results.

4.4 Discussion

There are many factors to consider, when determining if a generated response to a conversation is good or not: *Is the generated response grammatically correct? Does it respond to the previous utterance? Does it make sense in the context of the past conversation?* Giving scores for a generated response based on these questions is quite challenging. Additionally, in the case of CIMA, where the generation should be based on action labels, like *Hint* or *Correction*, the quality of a generated response is also determined by how well it matches the given label.

Unfortunately, there is currently no good way to evaluate the quality of a response based on these questions, except for human labeling, which is time-consuming and may be expensive. As mentioned in Section 4.1, we decided to use the BLEU score by Papineni et al. [2002] as a metric, which compares the target sequence with the generated one. This a good metric, but has a few weaknesses.

Sequence Length Tables 5 and 6 show two examples taken from the test set for CIMA. In Table 5, the generated response has a good score of 0.528, but when we look at the response, it is clear that it does not answer the student's question. The student asks for the Italian word for "tree" but gets the word for "blue". In comparison to that, the generated sequence in Table 6 accurately answers the question but only has a BLEU score of 0.049.

This happens because a single wrong word or missing words have much more impact on the score in shorter sequences than in longer ones. A human, however, would in this case probably give a lower score for the first example and a perfect score for the second one.

Context Keeping track of the past conversation and producing sequences which make sense in the given context is a general challenge when it comes to conversational systems. In Table 5, we see that the model has correctly learned the structure of the target sequence, i.e. *"[Word] is [Word]. Do you remember the word for [Word]?"*. It accurately generates the structure during generation but

Input	Context: Question, Hint, e vicino, viola, all albero, is next to the, purple, tree Tutor: can you try filling in the blank to the best of your ability ? Student: how do you say tree again ?
Target	tree is either all albero or l albera . do you remember the word for cat ?
Generated	the word for blue is well blu . can you remember word for tree now ?
BLEU	0.528

Table 5: The generated sentence gets a high score, even though the student's question was not answered, because in longer sentences, one wrong word has less weight. The sentence is also not consistent with the context since it wrongly introduces the color blue.

Input	Context: Hint, e di fronte al, verde, letto, is in front of the, green, bed Tutor: ok is in front of the is e di fronte al Student: okay ! but how do you say green ?
Target	green is verde .
Generated	verde
BLEU	0.049

Table 6: Even though the question is answered correctly, it gets a low score, because the generated sentence is shorter than the target.

Input	Context: Confirmation, e di fronte al, rosso, coniglio, is in front of the, red, bunny Tutor: remember that is in front of the is e di fronte al Student: how about il gatto e di fronte al coniglio rosso
Target	that s correct
Generated	yes
BLEU	0.0

Table 7: The target and the generated sequences are synonyms, but the score is low because they are not the same.

Input	Context: Hint, e di fronte al, viola, letto, is in front of the, purple, bed Tutor: please try to fill in the blank in italian . Student: il coniglio e dento scavalo rosa
Target	e di fronte al
Generated	bed is letto
BLEU	0.0

Table 8: The generated sequence matches the label and the context and is therefore a valid response. But since it is not the same as the target, it gets 0.0 points.

uses out-of-context words. It introduces a new color "blue" which can neither be found in the past conversation nor in the given context (which contains "purple"). This should be reflected better in the score.

Several possibilities In Table 7, we see the issue with synonyms. The phrases "that's correct" and "yes" are both confirmatory expressions which can be used interchangeably. But the rigid nature of the evaluation process gives it a score of 0.0. In the case of the example in Table 8, the generated phrase is not a synonym, but also an adequate response, which match the context. Here again, the received score is 0.0, but a human would probably give it a higher score.

Action Labels How well a generated response matches the action labels is very difficult to determine automatically. Again, the best current approach is human labelling. The Tables 5 and 7 show that the generated responses perfectly match the label of *Question & Hint* and *Confirmation*, respectively. In an ideal evaluation method, this should be reflected in the score. Overall, by looking at the generation results in the test set, we find that the generated sequences very often match the label. This was also found by Stasaski et al. [2020].

5 Related Work

Conversational datasets: There has recently been a lot of work to create large conversational datasets. For example Wang et al. [2019] use a concrete scenario to create conversations with focus on persuasion. The scenario is a role play between two people, where one tries to persuade the other to donate to charity. The idea is also to analyse and annotate which kind of persuasion tactics are used. In an educational setting, these different types of strategies can be translated to teaching strategies, like giving hints and asking questions among others.

Another dataset which concerns itself with the understanding of feeling was introduced by Rashkin et al. [2019]. Here the focus is on understanding what kind of emotion underlies a certain utterance. An example sentence of "I've been hearing some strange noises around the house at night." is then associated with emotions of fear, anxiety and worry. The extracted emotion can then be used to create a empathetic and appropriate response.

Educational focus: The Teacher-Student Chatroom Corpus (TSCC) by Caines et al. [2020] mentioned in Section 1 is a large dataset consisting of English lessons in form of conversations that were captured in an online chatroom. The conversations are authentic and with a clear educational goal of teaching English to the student.

Generation conditioned on labels: In the method mentioned above, responses are generated based on emotions. This idea can be extended to generating based on any kind of labels, or to control style aspects of the generated text.

Ficler and Goldberg [2017] introduce a way to not only control the generated content but also several stylistic aspects. The task is to generate a film review based on inputs about the style (e.g. professional vs. personal), the content (e.g. if the movie was perceived as good or bad) and some other labels. This provides a more fine-grained control over the generated reviews.

And lastly Smith et al. [2020] show an overview of different methods on how to control style aspects.

These methods could also be used in an educational settings, with different input parameters.

6 Conclusion

This thesis introduced a new conversational dataset WHJ, containing tutor-student conversations collected from the online learning platform White Hat Jr[10]. Since the conversations revolve around

[10]https://www.whitehatjr.com/

a concrete project, the project description, the student's code and the solution code are also added to the dataset. The WHJ dataset was cleaned by introducing labels to replace names, urls, E-mail addresses and phone numbers. This increased the performance notably. Another step that can be done in future to reduce noise and possibly improve performance is to correct spelling mistakes.

The pre-trained models BART and GPT-2 were fine-tuned on WHJ and the CIMA dataset and generated utterances given a conversation context. However, these models did not produce good results and the model often defaulted to generating very general responses. Possible reasons for this may be that the datasets CIMA and WHJ were too small. Also the way that the data was prepared which resulted in input sequences from the same conversation having the same prefixes, might have been problematic.

Our seq2seq model produced better results for WHJ and CIMA, obtaining the best BLEU score of 0.209 for WHJ and 0.328 for CIMA, which is slighty better than the best score Stasaski et al. [2020] achieve.

References

Katherine Stasaski, Kimberly Kao, and Marti Hearst. Cima: A large open access dialogue dataset for tutoring. pages 52–64, 01 2020. doi: 10.18653/v1/2020.bea-1.5.

Joseph Weizenbaum. Wiezenbaum, j.: Eliza - a computer program for the study of natural language communication between man and machine. communications of the acm 9(1), 36-45. *Commun. ACM*, 9:36–45, 01 1966. doi: 10.1145/365153.365168.

Sherry Ruan, Liwei Jiang, Qianyao Xu, Zhiyuan Liu, Glenn Davis, Emma Brunskill, and James Landay. Englishbot: An ai-powered conversational system for second language learning. pages 434–444, 04 2021. doi: 10.1145/3397481.3450648.

Andrew Caines, Helen Yannakoudakis, Helena Edmondson, Helen Allen, Pascual Pérez-Paredes, Bill Byrne, and Paula Buttery. The teacher-student chatroom corpus. pages 10–20, 11 2020. doi: 10.3384/ecp2017510.

Arthur Graesser, Shulan Lu, G. Jackson, Heather Mitchell, Mathew Ventura, Andrew Olney, and Max Louwerse. Autotutor: a tutor with dialogue in natural language. *Behavior Research Methods*, 36:180–192, 06 2004. doi: 10.3758/BF03195563.

Xavier Sánchez Díaz, Gilberto Ayala-Bastidas, Pedro Fonseca, and Leonardo Garrido. A knowledge-based methodology for building a conversational chatbot as an intelligent tutor. 09 2018.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. pages 7871–7880, 01 2020. doi: 10.18653/v1/2020.acl-main.703.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language Models are Unsupervised Multitask Learners. 2019. URL `https://openai.com/blog/better-language-models/`.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9:1735–80, 12 1997. doi: 10.1162/neco.1997.9.8.1735.

Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. volume 14, pages 1532–1543, 01 2014. doi: 10.3115/v1/D14-1162.

Tomas Mikolov, G.s Corrado, Kai Chen, and Jeffrey Dean. Efficient estimation of word representations in vector space. pages 1–12, 01 2013.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. 10 2018.

Dzmitry Bahdanau, Kyunghyun Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. *ArXiv*, 1409, 09 2014.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei Jing Zhu. Bleu: a method for automatic evaluation of machine translation. 10 2002. doi: 10.3115/1073083.1073135.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. Multilingual denoising pre-training for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:726–742, 11 2020. doi: 10.1162/tacl$_{a}$0343.

Xuewei Wang, Weiyan Shi, Richard Kim, Yoo Jung Oh, Sijia Yang, Jingwen Zhang, and Zhou Yu. Persuasion for good: Towards a personalized persuasive dialogue system for social good. 06 2019.

Hannah Rashkin, Eric Smith, Margaret Li, and Y-Lan Boureau. Towards empathetic open-domain conversation models: A new benchmark and dataset. pages 5370–5381, 01 2019. doi: 10.18653/v1/P19-1534.

Jessica Ficler and Yoav Goldberg. Controlling linguistic style aspects in neural language generation. pages 94–104, 01 2017. doi: 10.18653/v1/W17-4912.

Eric Smith, Diana Gonzalez-Rico, Emily Dinan, and Y-Lan Boureau. Controlling style in generated dialogue. 09 2020.

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Declaration of originality

The signed declaration of originality is a component of every semester paper, Bachelor's thesis, Master's thesis and any other degree paper undertaken during the course of studies, including the respective electronic versions.

Lecturers may also require a declaration of originality for other written papers compiled for their courses.

I hereby confirm that I am the sole author of the written work here enclosed and that I have compiled it in my own words. Parts excepted are corrections of form and content by the supervisor.

Title of work (in block letters):

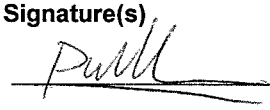
A Dialogue Tutoring Corpus and Agent for Teaching Programming

Authored by (in block letters):
For papers written by groups the names of all authors are required.

Name(s):	**First name(s):**
Puthenkalam	Christina

With my signature I confirm that
- I have committed none of the forms of plagiarism described in the 'Citation etiquette' information sheet.
- I have documented all methods, data and processes truthfully.
- I have not manipulated any data.
- I have mentioned all persons who were significant facilitators of the work.

I am aware that the work may be screened electronically for plagiarism.

Place, date	**Signature(s)**
Zürich, 03.06.2021	

For papers written by groups the names of all authors are required. Their signatures collectively guarantee the entire content of the written paper.