Conversational Agents Embodying a Character Using Neural Networks

Mihai Ilie and Traian Rebedea
University Politehnica of Bucharest, Faculty of Automatic Control and Computers
Bucharest, Romania
ilie.mihai92@gmail.com, traian.rebedea@cs.pub.ro

ABSTRACT
This paper presents a preliminary study on how a deep neural architecture can be used to build a chatbot which simulates a conversation having the style of a given character, including famous personalities. Thus, we analyze how well a sequence to sequence (seq2seq) neural model can learn to generate meaningful responses for open domain dialogue and to imitate a persona. For the experiments, we trained the model to imitate two different personas: Romania's most famous poet, Mihai Eminescu, and Eric Cartman from the South Park series. The model was trained without using any handcrafted rules on an end-to-end dialog and is able to build new utterances to a wide number of user inputs using original distinguished expressions for both characters. However, the generated utterances still need to be improved for a meaningful dialogue.

Author Keywords
Conversational agent, Dialogue systems, Word embeddings, Deep Neural Networks, Text style transfer

ACM Classification Keywords
I.2.7 Natural Language Processing; H.5.m HCI, Miscellaneous

INTRODUCTION

Conversational agents, also named chatbots when discussing about agents responding in a chat dialogue, have always been a topic of interest at the border of Natural Language Processing (NLP) and Computer-Human Interaction (CHI). They have been envisioned as an alternative, more natural way of interaction with a computer application using natural language and dialogue. Due to the recent advances in deep neural networks, chatbots have recently become again a hot topic, especially in NLP. Moreover, conversational agents have always been linked to the Turing Test, which is one of the most challenging problems in computer science research: can we train/build machines to act like humans in a conversation?

When developing a conversational agent there are several common challenges that sometimes are not so obvious. For example, retrieval-based models for chatbots use predefined responses and pick one of them using a set of heuristics, which can be either rule-based using logic and regular expressions or machine learning approaches for determining the best answer for a given input from a predefined list of answers. These models don't make any grammatical mistakes, their answers are syntactically correct, but can't handle questions for whom they don't have a predefined response. One recent example for this type of system is Google's Smart Reply system recently developed for Gmail [1].

On the other hand, generative models can handle new cases because they don't rely on (any) predefined responses and generate their own response starting from the human utterance they have to respond to. These generative models are special because they can give the users the sensation that they are talking with a real human mainly due to the multitude and variety of possible replies. As there are no predefined answers, these models need to learn possible ways to build answers using a large collection of conversations. Thus they require large dialogue datasets for training and their flexibility is limited due to the fact that grammatical mistakes are more frequent, especially for long sentences, mainly because they do not use any underlying grammars when generating responses.

Many of the conversational agents built until recently have been aimed at solving specific tasks. However, at this moment there is a growing interest in developing open domain chatbots. In an open dialogue system, there isn't a well-defined purpose or intention for the conversation, therefore the user can drive the conversation on any topic he/she wants. This is closely related to Artificial General Intelligence [2] since the amount of knowledge and the number of topics needed in order to create an authentic conversation are practical infinite in this scenario.

At the opposite side, a close domain conversational agent is easier to develop using retrieval-based models since the agent has a predefined task, like booking a hotel room or buying flowers from a shop. In this case the number of possible inputs and outputs is limited and the system just needs to fulfill its specific purpose. In this case, the users are not expecting the chatbot to provide answers on any other topic than the one it was designed for.

The paper continues with a section that provides a brief motivation for our proposed approach for solving the inconsistency of a chatbot’s utterances by incorporating a specific linguistic style. The following section discusses the most important recent works on building a persona-based conversational agent by inserting speaker embeddings into neural conversational models at training, but also by using standard rule-based approaches. Then we describe our proposed method, focusing on dataset construction, the model, and the method used for training. The last two sections of the paper present preliminary results for the two case studies and concluding remarks.

MOTIVATION

In this paper we focused on developing an open domain chatbot using a generative model which should be able to provide answers with a linguistic style similar to a given character. We call this type of conversational agent a model with personality. Although this term might be an over-statement for the current results, but it is an ambition for the future. We chose to implement a neural network architecture based on the sequence-to-sequence (seq2seq) model [3] for generating the response of the chatbot starting from other sequences of text, such as the human input. Seq2seq models have recently been very successful in other NLP tasks, such as machine translation and speech recognition. This model consists of two layers of recurrent neural networks using Long Short-Term Memory (LSTM) cells, an encoder and a decoder [3]. The input sequence is fed to the encoder which generates a word embedding for the entire input and for each word individually using a fixed size embedding vector. This vector is then passed to the decoder layer which generates the response one word at a time. To allow the decoder to have a more informed access to the input sequence, an attention mechanism was later introduced to the standard seq2seq model [4].

There is a major challenge when developing a chatbot using neural networks for two main reasons: the lack of a coherent personality and the inability of the network to generate longer sentences which are correct syntactically. In this paper, we aim to solve the first problem. When the model generates a response, we consider that it is desired that the response is related to a certain human persona. Most of the models generate plausible responses for the previous utterance of the user, but these responses are not consistent over a longer conversation, as can be seen in Table 1.

The major cause for this issue is that the training dataset for the neural network generative model (e.g. seq2seq) uses conversations from a multitude of different users in various contexts. On one hand, it is important to have a large dataset with conversations from various contexts in order to be able to respond to as many user inputs as possible. However, this creates inconsistency for the utterances generated by the chatbot.

We have tested our proposed approach in two completely different contexts. First, we developed a chatbot which should use the linguistic style of Romania's most famous poet, Mihai Eminescu. Then we tried to see the results on the character of Eric Cartman from the South Park series. In order to model the chatbot to talk in the same style as a given character, we started by collecting published works of Mihai Eminescu and movie transcripts from South Park. We first trained our neural model on a large dialogue corpus, then refined the training on the specific character dialogues alone. We consider that after this process, the resulting embedding vectors incorporated some of the linguistic style of the character - and we call this a chatbot with a specific personality.

User: what is your name
Bot: pris
User: how old are you
Bot: eighteen
User: where do you live
Bot: in the closet
User: how many legs do you have
Bot: twenty eight thousand

Table 1. Sample conversation with baseline chatbot

RELATED WORK

Until recently it was believed that a conversational agent must use a large number of predefined rules in order to handle all/most the cases in which it could respond to an input from the user. These rules were usually defined using specific markup languages and using regular expressions, such as Artificial Intelligence Markup Language (AIML) [5]. Chatbots using AIML, like ALICE Bot, that mapped a user input to a rule which was then used to retrieve a predefined response proved that this approach is practically impossible to develop an open-domain chatbot. The generated responses feel artificial, not relevant for many user inputs, and repetitive.

Changing the rule based approach with a machine learning approach was a step further. The Google Smart Reply [1] was trained on a corpus of labeled utterance responses in order to determine which utterance has a simple short response. The model encoded utterances and responses from the dataset into vectors using word embeddings and then measured their similarity by computing the dot product of the vectors. A large value proved that the vectors were similar and that corresponded with a positive label that the response was relevant for the given utterance. Even though these models can handle only conversations in a closed domain (e.g. meeting requests) due to the training corpora, they are very popular.

There have been several works on developing a chatbot with personality using generative neural networks, most noticeable of them is Li et al. [6]. They proposed to generate the embedding vector of a persona, based on the real person who generated the training dataset. This vector is called the speaker embedding. Afterwards, the speaker embedding was injected into the decoder of a generative model at every time step during training. This is equivalent to forcing the model to remember information about a person when generating a response.

We used a different approach, starting from the work of Nguyen et al. [7] who propose to pretrain the word embeddings for the decoder using a large dataset and then to build character embeddings by further training the decoder embeddings on a smaller dataset containing only the responses of the embodied persona. We have used a similar approach in our work, both for Romanian and for English in two different domains: generating texts with the linguistic style of Romanian poet Mihai Eminescu and with the persona of the movie series character, Eric Cartman from South Park.

There have also been some works that focused on developing a conversational agent to provide responses about the life of a given historical personality [8, 9]. Although this may seem similar to the topic of this paper, it differs in the fact that the generated responses are not individualized with the linguistic style of each character. Rather, they generate responses based on a mix of topic based AIML rules extracted from Wikipedia pages [8] or using a combination of knowledge base text generation and retrieval-based models from Wikipedia [9]. Thus the generated utterances do not carry the linguistic style of the persona.

PROPOSED SOLUTION

One major problem when training a generative neural model is underfitting caused by the small size of the training set. To overcome this problem, we first trained a generative chatbot on existent English large dialogue corpus. After this model was successful, we applied the same steps on training a chatbot on our own corpus gathered for Romanian.

Dataset Construction

For English, we used the Cornell Movie-Dialogs Corpus by Danescu-Niculescu-Mizil and Lee [10]. As a short description, the corpus contains 220,579 conversational exchanges between 10,292 pairs of movie characters and involves 9,035 characters from 617 movies. The problem that appears when trying to embed a persona from this corpus is that conversations have different styles for each movie. This leads to an unrealistic persona for the chatbot which will provide inconsistent answers between turns. Thus, the model trained on this corpus generated plausible responses, but not consistent ones.

In order to add more authenticity in the conversation we added dialogs from the South Park movie series. We collected more than 70,000 lines of dialogue together with character information from the following dataset https://www.kaggle.com/tovarischsukhov/southparklines. Because each context has its own speaker now it is easy to extract lines for a certain character. This way we can make a clear distinction in the dialog between different movie characters.

Cornell Movie Dialog Corpus (English)	**South Park Corpus**	**Cartman Corpus**
220,579	75,135	10,342
Subtitles from Movies Corpus (Romanian)	**Eminescu Corpus**	**Eminescu letters Corpus**
1,441,313	106,202	3,835

Table 2. Number of dialogue pairs for the different corpora (English and Romanian)

At the end, we gathered over 10,000 pairs of question-response for the character of Eric Cartman. Due to the small size of this dataset, we were not able to use these replies alone for training the neural generative model.

For the construction of the Romanian dialogue dataset, first we started by collecting movie subtitles in Romanian. At the end, our dataset contained 1,441,313 conversational replies. As conversations could not be assigned to a specific movie character due to the nature of the data, we considered one line as a given utterance and the following one as the reply to the previous one (this is the way subtitle lines appear in a movie subtitle file).

For the construction of the Mihai Eminescu corpus, we collected texts from the prose, journalism and personal letters of the author. We could not make a clear boundary between Eminescu's replies and ones of other characters as in the previous corpus, so we extracted only the replies from his personal letters and used them as utterance-response pairs. The size of this datasets are presented in Table 2, together with similar information about all the corpora used by our paper.

One important thing to notice for the Eminescu dataset is that the pairs of question-response in our corpus are not always aligned. For this reason, the topic for some utterances may differ from the topic of the corresponding answers. This can lead to generating responses that are not in the same context with the question.

Model Description

We used a slightly modified seq2seq model from Tensorflow (https://www.tensorflow.org/). It is composed of an encoder and a decoder which can be viewed as two different language models. The encoder transforms the input sentence into an embedding vector. The decoder uses the embedding vector as a map and generates the response one word at a time.

The model used in the experiments has 3 layers composed of 256 Gated Recurrent Unit (GRU) cells. GRU cells have been preferred to the standard LSTM cells used by the seq2seq model as they are a bit simpler and thus require less training data.

In order to improve the results, we reversed the input into the encoder and we used an attention mechanism to allow the decoder to have more informed access to the input sequence. The original decoder was implemented using beam search in a greedy fashion. As the result for each step individually was affected by local optima and the generated response was not very relevant, we decided to keep the best *k* candidates and perform a beam search thus exploring more results and getting an output which is somehow closer to the global optimum.

We used padding and bucketing to solve the problem of variable length of input sentences. For example, we consider buckets of sizes [(5, 10), (10, 15), (20, 25), (40, 50)]. For example, if the length of the input sentence is 4 and the length of the output sentence is 7 we put this conversation pair into bucket (5, 10) and pad with unused words until we fill to the maximum bucket size (e.g. 5 for encoder and 10 for decoder).

For the word embeddings, which is the method of representing words in a low dimensional vector space used by the seq2seq model, we used a size of 256 both for encoder and decoder.

Training

In order to overcome the problem of the small datasets for a given character, we used a different method for training:

- In the first phase, we pretrained the seq2seq model on large conversational datasets for 45,000 steps for both characters (Eric Cartman in English and Mihai Eminescu in Romanian). The first dataset was composed from Cornell Movie Dialog and South Park datasets. The second one was composed from Romanian movie subtitles dataset and entire Mihai Eminescu datasets.
- In the second phase, we trained on the specific corpus from which we extracted the character lines for about 10,000 steps. Thus, we trained on the South Park dataset and on the entire Mihai Eminescu dataset. We consider that this step is important to fine-tune the word embeddings for the given context of the character.
- In the final phase, we concentrated our training only on the character specific lines for 5,000 steps. We used utterance-response pairs from Cartman corpus and created utterance-response pairs from Eminescu's personal letters. This last step actually embodies the character persona into the model, mainly by changing the language model of the decoder.

We used separate vocabularies for the encoder and the decoder. In order to create a vocabulary containing only the most frequent words, we plotted their frequency versus number of words and chose the vocabulary dimension that covers most of the area under the curve (see Figure 1). Any other word was replaced with a special token called *UNK* (unknown). If the number of unknown tokens is large, the model learns to output UNK more frequently than other words and thus the conversation is flawed. We chose the vocabulary size to be 40,000 for both the English and the Romanian datasets.

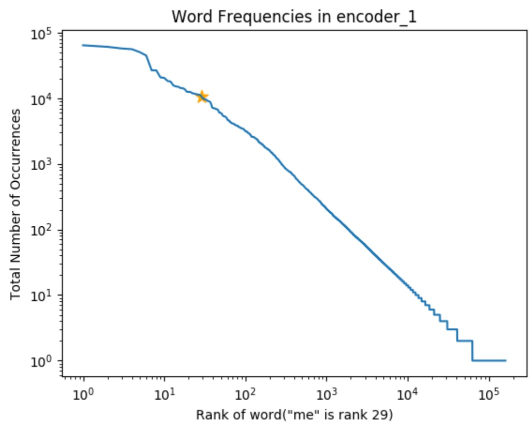

Figure 1. Word frequencies in the vocabulary of the encoder

RESULTS

As previously mentioned, we have tested our approach on two characters: Eric Cartman in English and Mihai Eminescu in Romanian. The models were able to embed some of the linguistic style (personality) of the two characters. The two conversational agents speak in the specific style of each character as can be seen in Tables 3 and 4. As there are no proper metrics for the evaluation of conversational agents, we used both human evaluation and metrics used in machine translation by comparing the generated utterance with the one issued by the character in the dataset (given a specific input from the dataset).

User: what is your name
Cartman-Bot: ohhhh kyle ohhhh mom
User: who are you
Cartman-Bot: you have to be kidding me
User: where do you want to go
Cartman-Bot: what does that mean i will just have to go back to school
User: who is your best friend
Cartman-Bot: ohhhh my god they killed kenny

Table 3. Sample conversation with Eric Cartman-based chatbot

As a side-note, we trained the Eminescu chatbot encoder with sentences also used by the decoder. As a consequence, instead of testing with normal sentences we had to use sentences that were similar with those from his letters.

If we compare the responses from the baseline chatbot with the responses from the Cartman chatbot we observe that the first ones are fitting better in the conversational context of the corresponding questions, but they lack the consistency of a personality. On the other hand, the responses from the Cartman chatbot do not always seem to be in the same context as the corresponding questions, but they incorporate the writing style of the character (see Table 3).

One possible reason might be that the baseline chatbot was trained on Cornell Movie Dialogue corpus which has better aligned pairs of question-response compared to our Cartman training dataset. This means that if we construct our dataset by supervising the correlation between questions and answer, we can improve the quality of the generated responses.

User: te sarut de mii de ori
Eminescu-Bot: al tau pentru totdeauna eminescu
User: ingerul meu iubit
Eminescu-Bot: UNK cu care eu n am fost niciodata
User: am suferit atat de mult pentru iubire
Eminescu-Bot: ma UNK ce as dori a ma ucide
User: ma pierd in negura timpului pierdut
Eminescu-Bot: din ceea ce nu voi sa se stie niciodata

Table 4. Sample conversation with Mihai Eminescu-based chatbot

Comparing Tables 3 and 4, we observe that Cartman responds to more general questions than Eminescu. This is because its encoder was trained on a more general dataset compared with the Romanian dataset for Eminescu. Moreover, the Eminescu chatbot generates more often UNK tokens than the Cartman one, showing that we need to have different vocabulary sizes for the two datasets.

In the future, in order to add diversity we will have to build a more general corpus for Eminescu by adding a new corpus with general utterances and align them with our corpus. After we can train the encoder only on the new corpus added and the decoder on the Eminescu corpus. This way we can test the Eminescu chatbot using normal sentences and the decoder will generate sentences having the desired style of the famous Romanian author.

Evaluation

The purpose of the performed evaluation is to measure that the conversational agent fulfill its task for a specific dialog. For open domain models there is no specific goal in the conversation. In this case, metrics like BLEU score which are used for machine translation are not perfectly suited because they are based on word matching and they show no correlation with human judgments as shown in [11]. The BLEU score measures the number of n-grams that appear in both in the candidate and the target utterance and thus represents the similarity between them. A strong similarity is correlated with a BLEU score closer to 1 and a low similarity with a score closer to 0.

We measured the BLEU score between the chatbot output and the actual response from the datasets for a given context utterance. As can be seen in Figure 2, there were very few similarities between the generated response and the actual response from the dataset. This is the reason why when testing with BLEU the scores are very low.

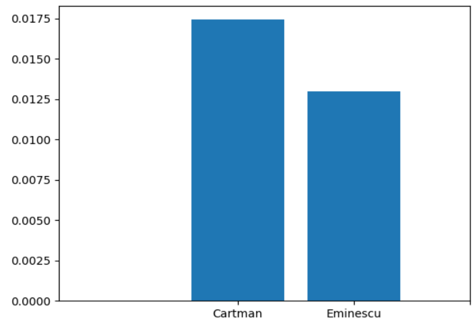

Figure 2. BLEU score for Cartman and Eminescu chatbots.

As an alternative evaluation, we designed an empirical evaluation using 10 human subjects and we used it as a metric to rate our model. This metric showed a higher score than the previous BLEU metric for the generated responses. For testing we picked 20 utterances that were not in the training corpus and we labelled them with 2 responses: the actual response from the South Park movie series and the Eric Cartman chatbot's output. We scored the generated responses that were tagged as responses from the movie series. The model received a score rate of 15%.

The human testers provided us with qualitative insights of our model. We observed that the chatbot repeated certain words that were not fit to the context in most of the generated sentences. This is the reason why most testers picked those sentences as being generated by the chatbot. We also observed that when generating longer sentences the chatbot was prone to grammatical errors or generated sentences that were not coherent. We think that this is caused by the maximum likelihood estimation (MLE) which finds the values for the parameters of our model which maximize the likelihood of an observation given those parameters. That is the reason why the responses contained words which had a high probability, words that were prone to appear many times in the training datasets.

Although the score received from the human testers is better than BLEU, the results can still be improved. This motivates us to enhance our model to achieve a higher score with human evaluation in the future.

As a last note, the fact that we could not properly evaluate our model implies that we also could not optimize it. As a result we could not tell if our model could have been improved or not if we had increased the number of training steps or changed the parameters for the seq2seq model.

CONCLUSIONS

This paper presented a method for building a chatbot using generative neural networks for determining the response in an open domain system that is able to answer using the writing style of a given human character. Thus, we have proposed to embed the personality (writing style, in this case) of the human character into a chatbot by modifying the training phase of the neural network. Slowly reducing the size of the datasets during the 3 phases of training by keeping only the character lines proved to be successful.

For future improvements we will concentrate on making the conversational agent able to remember information from the past utterances in the conversation by using an additional embedding for encoding recent past input sentences. More, we aim to improve the human evaluation score obtained by implementing a different architecture using Generative Adversarial Networks [12]. This architecture consist of two models: a generator (which is a seq2seq model) and a discriminator that classify utterances as human or machine generated. The discriminator is equivalent to the human testers in the Turing Test. The generator produces sentences and tries to fool the discriminator into believing that the sentences are generated by humans. The goal of this model is to generate utterances that are indistinguishable from human dialogue. Moreover, in this case we propose to use reinforcement learning to train the generator and the discriminator in a similar way to [13].

We will try to make a final improvement by adopting a Bayesian approach in the maximum likelihood framework as suggested in [14]. This way we will add two advantages: more expressiveness by an explicit representation of uncertainty and eliminating overfitting by regularization. This will also eliminate the problem with optimizing our model.

The adoption of a different training method in different phases was the key approach that gave a major breakthrough in personality of chatbots using neural networks. Future work in these directions will be developed.

ACKNOWLEDGMENTS

This research was partially supported by University Politehnica of Bucharest through the Excellence Research Grants Program UPB–GEX 13/30.09.2016.

REFERENCES

1. Kannan, A., Kurach, K., Ravi, S., Kaufmann, T., Tomkins, A., Miklos, B., ... & Ramavajjala, V. (2016). Smart reply: Automated response suggestion for email. *arXiv preprint arXiv:1606.04870*.
2. Goertzel, B. (2007). *Artificial general intelligence* (Vol. 2). C. Pennachin (Ed.). New York: Springer.
3. Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to sequence learning with neural networks. In *Advances in neural information processing systems* (pp. 3104-3112).
4. Chorowski, J. K., Bahdanau, D., Serdyuk, D., Cho, K., & Bengio, Y. (2015). Attention-based models for speech recognition. In *Advances in Neural Information Processing Systems* (pp. 577-585).
5. Wallace, R. S. (2009). The anatomy of ALICE. *Parsing the Turing Test*, 181-210.
6. Li, J., Galley, M., Brockett, C., Spithourakis, G. P., Gao, J., & Dolan, B. (2016). A persona-based neural conversation model. *arXiv preprint arXiv:1603.06155*.
7. Nguyen, H., Morales. D., & Chin, T. (2016). A Neural Chatbot with Personality. Technical Report, *Stanford University*.
8. Haller, E., & Rebedea, T. (2013). Designing a chat-bot that simulates an historical figure. In *Control Systems and Computer Science (CSCS), 2013 19th International Conference on* (pp. 582-589). IEEE.
9. Bogatu, A., Rotarescu, D., Rebedea, T., & Ruseti, S. (2015). Conversational Agent that Models a Historical Personality. In *Romanian Conference on Computer-Human Interaction (RoCHI) 2015*. Matrix Rom (pp. 81-86).
10. Danescu-Niculescu-Mizil, C., & Lee, L. (2011). Chameleons in imagined conversations: A new approach to understanding coordination of linguistic style in dialogs. In *Proceedings of the 2nd Workshop on Cognitive Modeling and Computational Linguistics* (pp. 76-87). Association for Computational Linguistics.
11. Liu, C. W., Lowe, R., Serban, I. V., Noseworthy, M., Charlin, L., & Pineau, J. (2016). How NOT to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation. *arXiv preprint arXiv:1603.08023*.
12. Li, J., Monroe, W., Shi, T., Ritter, A., & Jurafsky, D. (2017). Adversarial learning for neural dialogue generation. *arXiv preprint arXiv:1701.06547*.
13. Li, J., Monroe, W., Ritter, A., Galley, M., Gao, J., & Jurafsky, D. (2016). Deep reinforcement learning for dialogue generation. *arXiv preprint arXiv:1606.01541*.
14. Fortunato, M., Blundell, C., & Vinyals, O. (2017). Bayesian Recurrent Neural Networks. *arXiv preprint arXiv:1704.02798.*