

Aligning a medium-size GPT model in English to a small closed domain in Spanish

Alineando un modelo mediano GPT en inglés a un dominio cerrado y pequeño en español

Oscar Navarrate-Parra, Víctor Uc-Cetina, Jorge Reyes-Magaña
Universidad Autónoma de Yucatán, Facultad de Matemáticas, México
oscarranavpa@gmail.com, {uccetina, jorge.reyes}@correo.uady.mx

Abstract: In this paper, we propose a methodology to align a medium-sized GPT model, originally trained in English for an open domain, to a small closed domain in Spanish. The application for which the model is finely tuned is the question answering task. To achieve this we also needed to train and implement another neural network (which we called the reward model) that could score and determine whether an answer is appropriate for a given question. This component served to improve the decoding and generation of the answers of the system. Numerical metrics such as BLEU and perplexity were used to evaluate the model, and human judgment was also used to compare the decoding technique with others. Finally, the results favored the proposed method, and it was determined that it is feasible to use a reward model to align the generation of responses.

Keywords: Model alignment, GPT, reward model, human preferences.

Resumen: En este artículo se propone una metodología para alinear un modelo GPT de tamaño mediano, entrenado originalmente en inglés y de dominio abierto, a un dominio cerrado y pequeño en español. La aplicación para la cual se afina el modelo es para una tarea de preguntas y respuestas. Para lograr este objetivo también fue necesario entrenar e implementar otra red neuronal (a la cual llamamos modelo de recompensas) que pudiera calificar y determinar si una respuesta es adecuada para una determinada pregunta. Este componente sirvió para mejorar la decodificación y generación de las respuestas del sistema. Para la evaluación del modelo se utilizaron métricas numéricas como BLEU y perplejidad, y también se utilizó la evaluación a juicio humano, comparando la técnica de decodificación con otras. Finalmente, los resultados favorecieron el método propuesto, y se determinó que es factible utilizar un modelo de recompensas para alinear la generación de respuestas.
Palabras clave: Alineación de modelo, GPT, modelo de recompensas, preferencias humanas.

1 Introduction

Transformer neural networks have shown great potential for natural language processing (Vaswani et al., 2017). Several pre-trained transformer language models have been developed with the use of transfer learning, for instance the work by Radford et al. (Radford et al., 2018). These models have been scaled by increasing their number of parameters by hundreds of millions like GPT-

2 (Radford et al., 2019) and BERT (Devlin et al., 2019), or up to hundreds of billions like GPT-3 (Brown et al., 2020) and GPT-4 (OpenAI, 2023). Transformer models are trained with massive amounts of text and once they have been finely aligned, they can even recognize the task they must perform, despite never having been specifically trained for it. It is also possible to use the weights of these models and fine-tune them for a partic-

ular task, for example, open domain conversational systems like DialoGPT (Zhang et al., 2020), InstructGPT and ChatGPT (Ouyang et al., 2022).

On one hand, training huge models like GPT-4 is complicated and highly computationally expensive. On the other hand, medium-size models still lack coherence and sometimes tend to hallucinate facts. Furthermore, since they are usually trained with a wide variety of texts on the Internet, their behavior is not aligned to correctly follow the user’s instructions. This highlights the need for some additional component capable of providing consistency to the dialogue state, as well as some kind of system that evaluates and determines the quality of the responses (McTear, 2020). However, it is not always possible to have a large enough labeled dataset to fine-tune these models. Especially if the data comes from a knowledge base about a particular product, service or subject. Therefore, there is a need to search for feasible strategies for training language models in the response-generation task, that can be customized for a specific information domain.

In this article, we focus on the problem of aligning the DialoGPT model, to enable it to answer questions in Spanish. DialoGPT is a model based on GPT-2 and it was originally trained in English. In order to achieve a good alignment, the model is firstly fine-tuned with a corpus in Spanish. Then, it is further refined with a small collection of questions and answers about a specific topic. Finally, the performance of the model is improved through a reward neural network trained with human feedback. As a case study, we implemented a chatbot that answers frequently asked questions from a University undergraduate program, which we will refer to as the LCC dataset. The results show that it is feasible to use a reward model to align a medium-size GPT model to a small set of question-answer pairs.

2 Rewards based on human preferences

When modeling the environment for a reinforcement learning algorithm, a common practice is to manually design the reward function $R : S \rightarrow \mathfrak{R}$. However, for language processing, it is not as straightforward to design a reward system by hand. This is be-

cause the quality and accuracy of the generated sentences do not depend solely on their spelling and grammar, but also on their semantics and consistency with the query or assigned instruction. Therefore, to refine a language model through reinforcement learning, we need to employ a reward function that can distinguish the quality of the generated sentences, just as a person would. One way to achieve this is by training a reward model based on human preferences, that is, to train a sentence classifier in a supervised manner where the dataset is created from examples previously selected and labeled by human judges. Ziegler et al. (Ziegler et al., 2019) used this approach to fine-tune GPT-2, so that it could generate more realistic summaries. In order to do so, they represented the probability distribution of their model as a policy $\pi : S \times A \rightarrow [0, 1]$, and trained a reward model using human taggers, choosing the best tag from a small set of possibilities, to subsequently train the policy with the reward function based on human preferences – i.e. the rewards model learns to rate sentences like a human. One of the advantages of fine-tuning a language model with reinforcement learning instead of supervised learning is that such fine adjustment of the model requires less labeled data.

All of this suggests that it is possible to fine-tune pre-trained, not-too-giant language models by combining reinforcement learning techniques and human preferences, obtaining models capable of understanding a wider range of language and responding more accurately to user queries. In other words, language models can be practically aligned using reinforcement learning components such as a reward function.

3 Methodology

The proposed methodology is illustrated in Figure 1. In general we have the following components:

- Acquisition of a database in Spanish, which is used for the first fine-tuning of the DialoGPT conversational model.
- Acquisition of a small dataset about a particular topic, namely, the LCC dataset.
- Refinement of the conversational model using the LCC dataset.

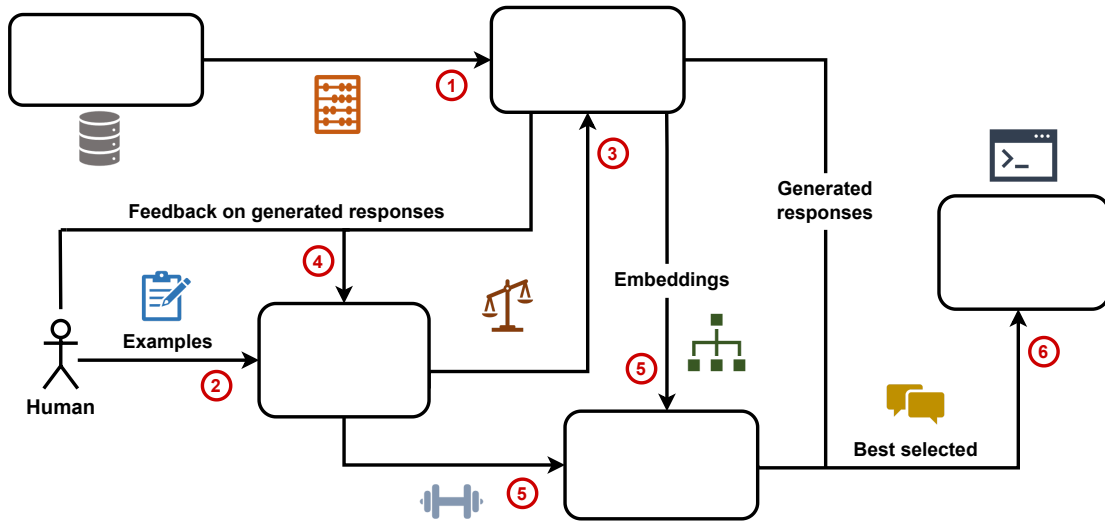


Figure 1: Outline of the methodology.

- Manual rating by humans of the new responses generated. The resulting rating is further used to augment the previous version of the dataset.
- The augmented dataset and the embeddings generated from the fine-tuned DialoGPT model are used to train a neural network that predicts rewards for each response, namely, the reward model.
- Optimization of the conversational system via a decoding algorithm with rewards in order to align it to the specific-topic dataset.

4 *Conversational model*

In this work we use the DialoGPT model because it is a tunable and relatively small language model originally trained to generate dialogues in English. Moreover, it is possible to further refine this model and adapt it to a different language (Adewumi et al., 2021).

Thanks to the fact that DialoGPT is a sizable model, it is possible to train it without the need for huge and sophisticated computing equipment, unlike larger models such as GPT-3 and GPT-4. In this case, the medium-size DialoGPT model trained from GPT-2 checkpoint was used, and we further fine-tuned it using the Spanish conversation database. In this way, GPT-2’s weights help the model to understand language structure while DialoGPT’s weights provide conversational insight. The same idea is followed to

adjust DialoGPT to the Spanish language, in order to further adapt it to the LCC dataset.

4.1 *Conversations dataset in Spanish*

To refine the model to the Spanish language, a subset of the original work database was translated. This dataset was obtained from the Reddit file server¹, specifically, we extracted the conversations between the years 2006 and 2007 of the site. The resulting dataset had a size of 379KB and consisted of 110 thousand examples. We used OPUS-MT (Tiedemann and Thottingal, 2020) to translate the dataset from English to Spanish, employing the one-way translator model via the EasyNMT library². Prior to translation, we split the dataset into an 80% training set and a 20% validation set.

5 *Reward model*

The responses that the refined DialoGPT model generates are in Spanish but lack coherence and are not aligned to the user’s intention. Furthermore, since the model was trained on a set of open domain conversations, the model does not really serve a useful purpose. To mitigate this problem, we used a reward model based on human preferences, in order to improve the quality of the responses generated by DialoGPT and align the model

¹File server address at <https://files.pushshift.io/reddit/>.

²EasyNMT Github repository at the link <https://github.com/UKPLab/EasyNMT>.

to a set of closed-domain dialogues focused on a particular topic.

5.1 Creation of the dataset from human preferences

Once the first fine-tuning on the Spanish dataset is completed, we proceeded to build the dataset from human preferences to further refine DialoGPT and to train a new reward model. This dataset was collected in two stages.

In the first stage, a human tagger creates a new set of (question, answer) pairs of a specific topic (i.e. the LCC dataset), generating between 50 and 100 examples, to slightly fine-tune the model. It is important to consider two factors during this stage. First, this additional fine-tuning of the model must not drastically alter the current values of the parameters, otherwise, all previous training could be forgotten and the model would be limited to generating responses only from this dataset. On the other hand, it is at this moment that the dataset to which we want the model to be aligned with is included. So, more examples of dialogues can be added to improve its social behavior.

In the second stage, once the model has been slightly fine-tuned, the human tagger rates the newly generated responses on a scale of 0 to 1, using the dataset created in the first stage. We save the tuples (question, response, rating) into a new dataset that also contains examples from the previous set. This newly generated dataset is then used to train a reward model.

5.2 Implementation of the reward model

The reward model is essentially a feed-forward neural network, and its architecture is described in Figure 2. The network takes as input, the embeddings of two sentences, the question and its respective answer. Since the embedding layer of the fine-tuned model already provides a good representation of the Spanish words, this step speeds up the learning process. These vectors are then passed to a smoothing layer and the resulting vector is processed through two layers with ReLU activation function. In the final layer, the softmax activation function converts the values to probabilities, which will be used to directly qualify a (question, answer) pair.

5.3 Optimization of the response generation

The algorithm for optimizing the responses using the reward model is given in Algorithm 1. First, we calculate N candidates for the next token to be generated, based on the probabilities obtained from the language model. Then, we calculate the score for each of these candidates, using the already trained reward model, which takes as input the embeddings of the input sequence and the embedding of each candidate. The reward model assigns a score to each candidate, which measures its semantic similarity to the input sequence. Finally, the candidate with the highest score is selected as the next token to be generated.

The embeddings of a sentence are calculated by adding all the embeddings of each token appearing in that sentence. This process results in a fixed-length vector representation of the sentence that captures its semantic content.

6 Experimental work

For the training of both the language and reward models, a computer with Intel(R) Xeon(R) 2.30GHz CPU, 12.7 GB RAM memory, Nvidia Tesla T4 GPU with 16GB GDDR5 1.59GHz and 107.7 GB of hard disk storage was used. In the first instance, the model was fine-tuned using the Spanish dataset, and later it was further refined using the LCC dataset. The hyperparameters used in these two training phases are shown in Table 1.

To prevent overfitting, we used regularization by penalty of L1 weights during training. In addition, we employed the Noam decay scheme (Vaswani et al., 2017) with a warmup of 200 steps, which dynamically adjusts the learning rate during training to improve convergence.

6.1 DialoGPT model training

Figure 3 shows the result of the first training. Notice how the model converges rapidly during the first few iterations, and then it starts to slow down. This is because the model was not tuned from scratch, and although it was trained in another language, the model already had basic knowledge about conversations and language. During the second training, the loss value of the model is very unstable at the beginning, which is likely due

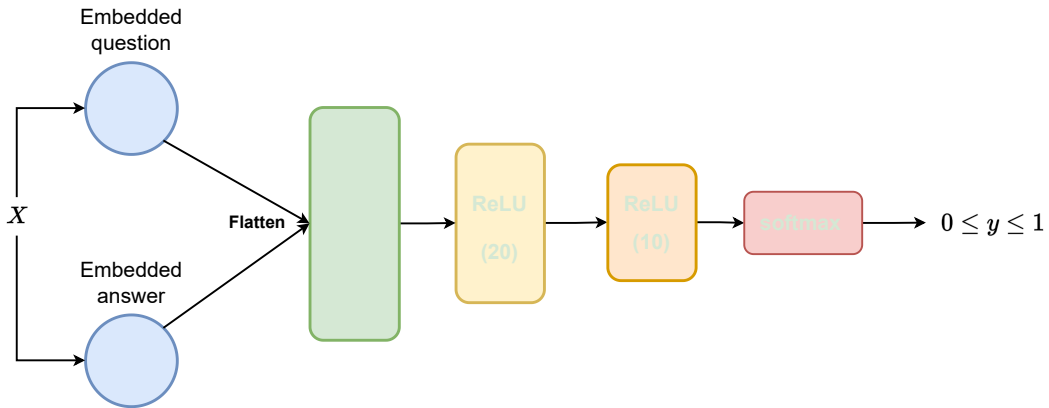


Figure 2: Neural network architecture for the reward function.

Algorithm 1: Decoding algorithm with rewards

```

Input :  $X$ : Input token sequence
Input :  $T$ : Temperature
Input :  $L$ : Maximum output length
Input :  $N$ : Number of candidates
Output:  $Y$ : Output text sentence

1  $Candidates \leftarrow []$ ;           /* candidates is initialized as an empty list */
2 for  $i \leftarrow 1$  to  $N$  do
3    $Y \leftarrow []$ ;
4   for  $j \leftarrow 1$  to  $L$  do
5      $logits \leftarrow Model(X)[-1]/T$ ;
6      $token \leftarrow multinomial(softmax(logits))$ ;
7      $Append(token, Y)$ ;
8     if  $token = Model.EOS$  then
9       Break;
10    end
11     $X \leftarrow X + token$ ;
12  end
13   $Append(Y, Candidates)$ ;       /* Y is added to the candidate list */
14 end
15  $Scores \leftarrow []$ ;
16 for  $i \leftarrow 1$  to  $N$  do
17    $c \leftarrow Reward(Embedding(X), Embedding(Candidates[i]))$ ;
18    $Append(c, Scores)$ ;
19 end
20  $m \leftarrow ArgMax(Scores)$ ;
21 return  $Tokenizer.decoder(Candidates[m])$ ;

```

to the fine-tuning of the model’s pre-existing parameters. However, once the first epoch is completed, the model converges extraordinarily quickly (see Figure 4). This may be due in part to the many-to-many structure of the LCC dataset, which contains a diverse range of questions and answers, allowing the model to learn more effectively from different

types of conversational data.

6.2 Reward model training

Using the LCC dataset and the model tuned with it, 5 answers were generated for each question using sampling. Then, each pair (question, answer) was labeled with a score in the interval $(0, 1)$. All 90 examples from

	Spanish model	Spanish model + LCC dataset
Batch size	64	1
Learning rate	1×10^{-5}	1×10^{-5}
Scheme	Noam decay	None
Epochs	2.5	2
Reg. factor	0.01	0.001
Time	4 hours	10 minutes

Table 1: Training hyperparameters.

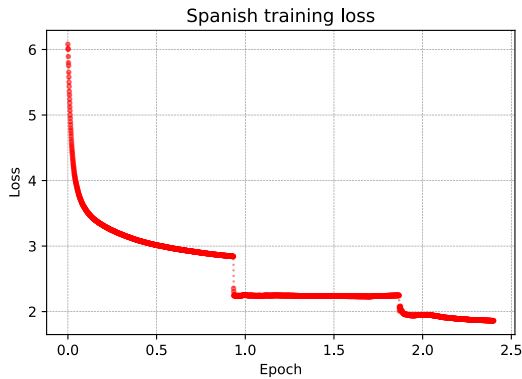


Figure 3: Graph of the cost function for training with the dataset in Spanish.

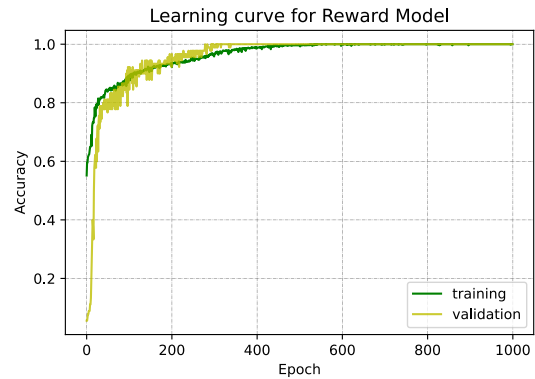


Figure 5: Reward model learning curve graph.

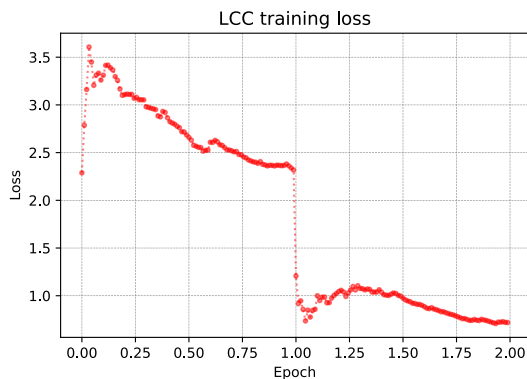


Figure 4: Plot of the cost function for training with the LCC dataset.

the previous LCC dataset were added to this new set, with a score of 1.0, making a total of 540 (question, answer, score) examples. The resulting dataset was used to train the reward model for 1000 epochs, with a learning rate of 0.001. To evaluate this model, the precision of the neural network was measured in two datasets: the training and the validation one; the latter is the original LCC dataset, and was used to verify that at least the learned reward function is capable of giving a positive score to all correct (question, answer) pairs. Figure 5 illustrates both learning curves and

shows that the model has learned to correctly reward all right answers for each question, however, it is still difficult to assess whether the model is capable of scoring all the wrong dialogues with a negative rating.

7 Evaluation of the generated responses

In order to evaluate the performance of the reward model, the generated responses were evaluated by comparing the system with 3 different decoding algorithms: filtered sampling with $\text{top-}k=20$ and $\text{top-}p=0.9$, sampling with the same filtering as above and using the rewards model with 10 candidates, and beam search with $b=6$; we choose a generation length of 200 tokens and temperature $T=1$.

For the automatic evaluation, the perplexity metric and the BLEU score (Papineni et al., 2002) were used. Perplexity measures how well the model’s output probability distributions predict the target tokens, therefore a low perplexity corresponds to a better performance. While BLEU analyzes if the generated sentences contain sets of occurrences that are also present in the training examples.

7.1 Results and discussion

Table 2 shows the results. Using the BLEU metric, normal sampling achieved a higher accumulated score on the 90 instances while reward sampling obtained almost 4 points less. Both methods should have received a score of at least 45 as almost half of the predicted responses were correct. However, this was not the case because BLEU automatically penalizes with 0 those very small N -grams such as uni-grams, bi-grams and tri-grams. Although the sampling technique was superior in this evaluation, this does not mean that it is better due to the lack of references and the deficiency of evaluating a conversational system by comparing N -grams, because unlike machine translation, a question can have too many associated answers.

On the other hand, in the evaluation of perplexity, sampling with rewards showed greater superiority, this means that the rewards model is capable of generating a better probability distribution than simple sampling. Furthermore, both techniques significantly outperformed the traditional beam search technique, which indicates that response generation is a more complex task than simply maximizing token sequence probabilities.

Likewise, an evaluation of human judgment was also carried out. That is, the answers that each technique generated through the review of a person were evaluated, considering characteristics of the answers such as coherence and relevance with respect to the question. The outcome was the preference towards the responses generated by the reward sampling technique. Some examples of conversations generated using reward sampling are shown in Table 3.

About 44% of the responses that the model generated using this technique were either identical to those of the original dataset or at least appropriate according to the user's intent. Another 32% of the responses did not provide true information but instead hallucinated facts or the model wasn't able to correctly interpret the query. The rest of the responses were basically gibberish incoherent sentences. Many of these examples were ridiculously complex explanations and made no semantic sense, while others tended to have many grammatical errors with nonsensical word repetitions and made-up conjugations. This is mostly because the model was

trained on a very small dataset. Language models typically tend to be trained with millions of instances to allow the model to fully capture the meaning of phrases and words. Another reason why the chatbot, even using the reward model, generates these types of responses may be due to the quality of translation from English to Spanish of the first dataset. In this case, it was not possible to manually review and correct the translation of each sentence to avoid this inconvenience. Despite this, the technique implemented with the reward model generated a larger number of correct responses and fewer gibberish responses, compared to sampling without rewards.

8 Conclusion

In this work, a pre-trained state-of-the-art transformer model for dialogue generation was implemented and refined through two consecutive fine-tuning phases in order to align it to a small dataset of questions and answers about one specific undergraduate program at our University. To this end we used two datasets. One dataset was used to fit the model to the target language and the second one was used to align it to the closed-domain dataset. For this second step, a separate neural network was also implemented and trained, so that it learns to qualify and reward conversations according to human preferences. This model was trained with feedback data from the conversational system. Finally, the reward model was used to optimize the dialogue generation of the tuned transformer and its performance was compared with two other response decoding techniques. The result was a decent improvement over the filtered sampling technique and a huge superiority over beam search. The results also show that it is feasible to align a medium-sized language model to a small dataset for a complex problem such as response generation.

Finally, it is important to mention that the use of a corpus originally generated in Spanish, and not an automatically translated one, as we did in this work for the first fine-tuning of the conversational system, would have certainly achieved a positive impact in the final performance. Also, the use of a more recent language model, like a larger GPT version, would improve the quality of the responses in the question-answer application,

	Sampling	Sampling with reward	Beam search
BLEU	20.70	16.99	7.82
Perp.	8.49	6.20	12.09

Table 2: Evaluation with numerical metrics.

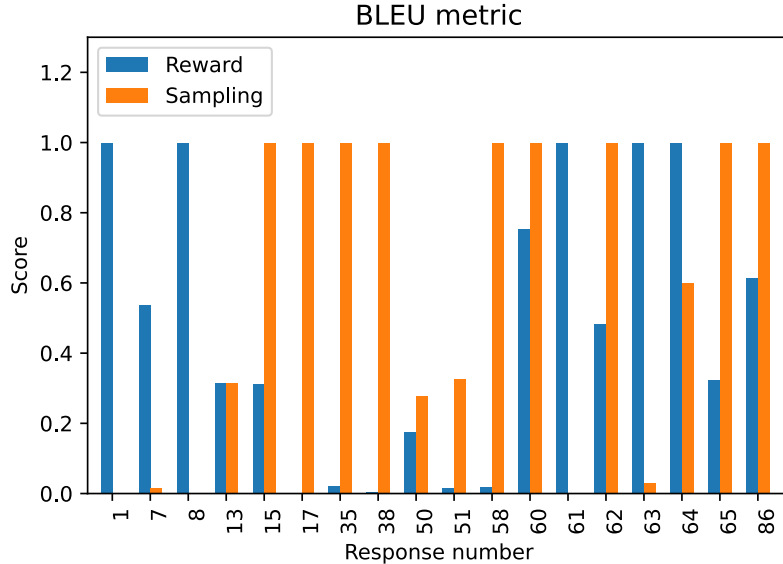


Figure 6: Evaluation using BLEU.

considering of course that retraining any of the newest data-hungry versions would impose some technical constraints, in terms of the computational power needed to do it.

Acknowledgments

We thank the reviewers of this article for their comments and suggestions.

References

- Adewumi, T., N. Abid, M. Pahlavan, R. Brännvall, S. S. Sabry, F. Liwicki, and M. Liwicki. 2021. Småprat: Dialogpt for natural language generation of swedish dialogue by transfer learning.
- Brown, T. B., B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei. 2020. Language models are few-shot learners.
- Devlin, J., M.-W. Chang, K. Lee, and K. Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.
- McTear, M. 2020. *Conversational AI: Dialogue Systems, Conversational Agents, and Chatbots*. Morgan & Claypool.
- OpenAI. 2023. Gpt-4 technical report.
- Ouyang, L., J. Wu, X. Jiang, D. Almeida, C. L. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, J. Schulman, J. Hilton, F. Kelton, L. Miller, M. Simens, A. Askell, P. Welinder, P. Christiano, J. Leike, and R. Lowe. 2022. Training language models to follow instructions with human feedback.
- Papineni, K., S. Roukos, T. Ward, and W.-J. Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA, July. Association for Computational Linguistics.
- Radford, A., K. Narasimhan, T. Salimans, and I. Sutskever. 2018. Improving language understanding by generative pre-training.

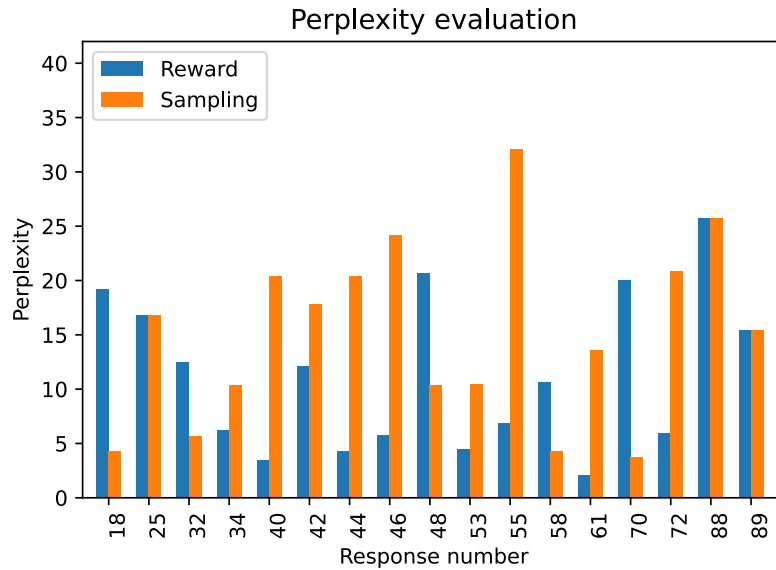


Figure 7: Evaluation using perplexity.

User	Chatbot
¿En qué áreas me puedo especializar?	Inteligencia Artificial, Cómputo Científico, Ciencias de Datos, Redes de Computadoras, Sistemas Operativos, Realidad Aumentada, entre otros.
¿Tengo que estudiar mucho?	En esta carrera se presentan las bases matemáticas que dan paso a las aplicaciones de la computación. También se da una introducción a la algoritmia y programación. Es cuestión de práctica. Si te gustan las matemáticas y lógica, la programación y los retos ¡te recomendamos Ciencias de la Computación!
¿Cuántos años se estudia?	El plan actual tiene una duración desde 9 semestres a 18 semestres.
¿Hay apoyos económicos?	Sí, hay varios tipos de apoyos y becas que puedes consultar en los sitios oficiales.
¿Qué lenguajes de programación aprenderé?	Te enseñamos desde lo básico y también varios tipos de lenguajes de programación. Aquí aprendemos a programar en lenguaje C, Java y Python.

Table 3: Examples of conversations using reward sampling.

Radford, A., J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever. 2019. Language models are unsupervised multitask learners.

Tiedemann, J. and S. Thottingal. 2020. OPUS-MT — Building open translation services for the World. In *Proceedings of the 22nd Annual Confereneec of the European Association for Machine Translation (EAMT)*, Lisbon, Portugal.

Vaswani, A., N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez,

L. Kaiser, and I. Polosukhin. 2017. Attention is all you need.

Zhang, Y., S. Sun, M. Galley, Y.-C. Chen, C. Brockett, X. Gao, J. Gao, J. Liu, and B. Dolan. 2020. Dialogpt: Large-scale generative pre-training for conversational response generation.

Ziegler, D. M., N. Stiennon, J. Wu, T. B. Brown, A. Radford, D. Amodei, P. Christiano, and G. Irving. 2019. Fine-tuning language models from human preferences.