EDUCACIÓN
SECRETARÍA DE EDUCACIÓN PÚBLICA

Instituto Politécnico Nacional
‘La Técnica al Servicio de la Patria”

# Research in Computing Science

Vol. 152 No. 12
December 2023

# Research in Computing Science

***Research in Computing Science***, Año 22, Volumen 152, No. 12, diciembre de 2023, es una publicación mensual, editada por el Instituto Politécnico Nacional, a través del Centro de Investigación en Computación. Av. Juan de Dios Bátiz S/N, Esq. Av. Miguel Othon de Mendizábal, Col. Nueva Industrial Vallejo, C.P. 07738, Ciudad de México, Tel. 57 29 60 00, ext. 56571. https://www.rcs.cic.ipn.mx. Editor responsable: Dr. Grigori Sidorov. Reserva de Derechos al Uso Exclusivo del Título No. 04-2019-082310242100-203. ISSN: en trámite, ambos otorgados por el Instituto Politécnico Nacional de Derecho de Autor. Responsable de la última actualización de este número: el Centro de Investigación en Computación, Dr. Grigori Sidorov, Av. Juan de Dios Bátiz S/N, Esq. Av. Miguel Othon de Mendizábal, Col. Nueva Industrial Vallejo, C.P. 07738. Fecha de última modificación 01 de diciembre de 2023.

Las opiniones expresadas por los autores no necesariamente reflejan la postura del editor de la publicación.

**Research in Computing Science,** year 22, Volume 152, No. 12, December 2023, is published monthly by the Center for Computing Research of IPN.

The opinions expressed by the authors does not necessarily reflect the editor’s posture.

# Advances in Artificial Intelligence

**Hiram Ponce (ed.)**

Instituto Politécnico Nacional, Centro de Investigación en Computación
México 2023

**ISSN: in process**

Formerly ISSNs: 1870-4069, 1665-9899

Instituto Politécnico Nacional (IPN)
Centro de Investigación en Computación (CIC)
Av. Juan de Dios Bátiz s/n esq. M. Othón de Mendizábal
Unidad Profesional "Adolfo López Mateos", Zacatenco
07738, México D.F., México

http://www.rcs.cic.ipn.mx
http://www.ipn.mx
http://www.cic.ipn.mx

The editors and the publisher of this journal have made their best effort in preparing this special issue, but make no warranty of any kind, expressed or implied, with regard to the information contained in this volume.

Indexed in LATINDEX, DBLP and Periodica

Electronic edition

ISSN 1870-4069

# Table of Contents

ISSN 1870-4069

# Information Retrieval Techniques for Question Answering based on Pre-Trained Language Models

Ángel Cadena[1,3], Francisco F. López-Ponce[2,3],
Gerardo Sierra[3], Jorge Lázaro[4],
Sergio-Luis Ojeda-Trueba[3]

[1] Universidad Nacional Autónoma de México,
Instituto de Investigaciones en
Matemáticas Aplicadas y Sistemas,
Mexico

[2] Universidad Nacional Autónoma de México,
Facultad de Ciencias,
Mexico

[3] Universidad Nacional Autónoma de México,
Insituto de Ingeniería,
Mexico

[4] Universidad Nacional Autónoma de México,
Escuela Nacional de Lenguas,
Lingüística y Traducción,
Mexico

francisco.lopez.ponce@ciencias.unam.mx

**Abstract.** This paper presents a comparative study between two prominent pre-trained language models, RoBERTa and GPT-3, focused on their performance in Question Answering (QA). Broker exams serve as a rigorous evaluation guide, with which we examine the effectiveness of these models in understanding complex questions based on newly presented information in the form of the 19 Code of Federal Regulations of the United States (19 CFR). Our findings reveal insights into the strengths and limitations of each model, shedding light on their suitability for specific QA applications in the finance and legal domain. RoBERTa offers a fast implementation of a QA model yet it struggles processing complex questions, whereas GPT-3 is able to answer efficiently a wide range of reason based questions.

**Keywords:** Question answering, large language models, broker exam.

## 1 Introduction

Question Answering (QA) is a branch of Natural Language Processing (NLP) that focuses on the development of a model capable of efficiently answering human generated questions based on certain available information.

pp. 5–15; rec. 2023-06-04; acc. 2023-09-01

There are two main approaches to this problem: extractive QA which focuses solely on retrieving the specific data from a text and generative QA that creates text based on the information most relevant to the query. With the development of the Transformer architecture QA has evolved from fully controlled descriptive systems to pre-trained models fine-tuned to this task. Task focused datasets such as the Stanford Question Answering Datasets (SQuAD) version 1.0 [9] and version 2.0 [8], or the MLQA [5] a 7 language QA dataset, offer benchmarks for these models.

Extractive models like BERT [3] achieved an 87.4% of exact matches (EM), and 93.2 F1-Score on SQuAD v1.0, whereas in the English section of MLQA achieved 67.4% EM and 80.2 F1. A multilingual counterpart like XLM-RoBERTa [2] fine-tuned on SQuAD outperforms BERT in the English section of MLQA obtaining 67.8% EM and 80.6 F1. On the other hand, generative models like GPT-3 [1] outperform both of their predecessors, obtaining a 90.7% EM and a 93.0 F1 on the more complex SQuAD v2.0. In this paper we compare the performance and the results of an encoder such as RoBERTa against a decoder such as GPT-3 in this context.

We will work with publicly available USA Customs laws as well as Broker exams, with which we will demonstrate the capabilities of each model in terms of Information Retrieval and QA based on different levels of reading comprehension and user generated queries. In section 2 we show the conceptual differences between encoders and decoders, with information regarding the models chosen for this experiments.

In section 3, we explain the use of customs related information as well as the computational handling of legal texts needed for each model. Section 4 cover practical implementations, results and evaluations, in section 5 we present analysis of the results and in section 6 we conclude the article and present future work.

## 2 Background

Pretrained language models can be differentiated into three types: encoders, encoders-decoders, or decoders, depending on their intended use as well as their training methods. Encoders such as BERT create representations of text data based on masked language modeling. This type of model predicts a hidden token given a sentence, having access to every token in that sentence.

Decoders and Generative models like GPT focus on predicting the next word in a sentence, hence they only have access to previous tokens and their training procedure works with self-supervised learning. In terms of adjusting a model to a particular NLP task, encoders need to be fine tuned with a dataset focused on the chosen task, whereas decoders can avoid this step and work on the task without updating the model's parameters.

QA focused datasets help the fine tuning process but still fall short when dealing with unseen information. To solve this problem we need to feed each model the precise information we need them to process. Considering the computational expenses of a robust fine tuning as well as the necessity of having an appropriate dataset, we aim to compare these two architectures in order to obtain a QA model independent of fine-tuning and an increased dataset. The two models chosen for this comparison are RoBERTa and GPT-3.

**Table 1.** Structure of the 19 CFR.

| Depth | Content | Reference |
|---|---|---|
| Chapter | U.S. Customs and Border Protection | I |
| Part | Special Classes of Merchandise | 12 |
| Section | Release under bond, liquidated damages. | 12.3 |
| Subsection | Bond amount | 12.3 (b) |
| Subindex and text | Three times the value of the merchandise as provided in 113.62(n)(1) | 12.3 (b)(2) |

### 2.1 RoBERTa

RoBERTa [6] is an improvement of BERT [3] based on four modifications: (1) Training the model longer, with bigger batches, and over more data; (2) removing the next sentence prediction objective; (3) training on longer sequences; and (4) dynamically changing the masking pattern applied to the training data. BERT [3] was trained in two objectives: Masked Language Modeling (MLM) and Next Sentence Prediction (NSP).

The aim of MLM involves that given an input sequence of known tokens with one randomly replaced with the special token [MASK], the model should predict the actual value of the masked token. Next Sequence Prediction is a binary classification loss for predicting whether two sentences follow each other in the original text. In modification (1) the authors consider five English-language corpora of varying sizes and domains totaling over 160GB of uncompressed text and batch sizing set to 8k and the training steps were increased to 100k, 300k and 500k.

Modifications (2) and (3) come from a series of experiments conducted by the authors where they decide to remove the NSP loss. Each input contains full sentences sampled contiguously from one or more documents, such that the total length is at most 512 tokens. Inputs may cross document boundaries. If the end of one document is reached sentences from the next document are sampled using a separator token between documents. A final modification is the dynamic masking. The original implementation of BERT performs masking once during data preprocessing which results in a single static mask. For RoBERTa was used an dynamic masking instead, where the masking pattern is generated every time a sequence is given to the model.

### 2.2 GPT-3

GPT-3 presented in [1], is an auto-regressive language model trained with in-context learning that aims to solve several NLP tasks without the need of fine tuning the model on a task specific dataset. Transformer language models have been shown to increase performance based on the amount of parameters used during training [4], leading to GPT-3 being trained with 175 billion parameters, a substantial amount compared to BERT's 110 million. GPT-3 is a decoder based language model that works by queries (also called prompts) given by an user.

The model processes the user's query and generates an adequate response, but the response varies based on the structure of the query. GPT-3 was trained using three types of queries: Zero-shot, where there's no example of the intended use of the query; One-shot, where the query shows just one example of the task; and Few-shot, where the query shows multiple examples of the task in hand. For our task we focus on testing GPT-3's Zero-shot behavior in the context of Question Answering, this means that user prompts given to the model will be solely the questions with which we aim to evaluate. The only adjustment made will be a system prompt that modifies GPT'3 behavior for all iterations.

## 3 Code of Federal Regulations

For the theoretical evaluation of the aforementioned models we decided to work with customs laws of the United States. Legal information presents a considerable amount of free, accessible, and structured data easily adaptable to a QA problem. In particular Customs laws in the United States are an adequate branch given the existence of Broker Exams with complex questions, deep theoretical context, and Answer Keys that serve as an evaluation guideline for brokers and, in our case for a pre-trained model.

Broker Exams are composed of 80 questions based on the contents of 5 different legal documents. These questions are divided into 15 sections based on their subject and the answers are presented in a multiple choice format. The Answer Key contains the correct answer for each question in the test and includes the name of the legal document and the section within needed to answer the corresponding question. More importantly these questions range from standard information extraction to problem solving based on an explicit example, making them a more complex frame of work since understanding the context is a necessity in order to answer correctly.

The most referenced document, with more than half of the questions referencing it, is the Code of Federal Regulations (CFR) in the Answer Key from the October 2022 exam. The CFR is a collection of laws about numerous subjects written and presented yearly by the US Government divided into 50 titles. Title 19, named Custom Duties [7], is the most referenced text in this Answer Key. There is available an XML version of each of its chapters. This make it the best option in terms of relevancy and computational performance. We will refer to this text as 19 CFR onward.

The 19 CFR is comprised of four chapters containing 600 parts, each part referring to a precise set of information, additionally each part is divided into sections that contain a numbered subsection and, even though it doesn't always happen, a sub-index. Table 1 describes the levels of depth with an example of Part 12, Section 12.3, Subsection 12.3 (b), Subindex 12.3 (b)(2), as well as the corresponding text. We worked only with Chapter I, which encompasses parts 1-199, since those parts are the only ones mentioned in the Answer Key.

We filtered out the answers that do not come from the 19 CFR as well as those which require calculations or were questions of exclusion, leaving 44 questions adequate for an information retrieval task. After further analysis of the Answer Key, we observed that almost 40% of the remaining questions can be answered using only sections 111 and 113, thus we focused on a qualitative evaluation of those 16 final questions.

**Table 2.** Answers and prompts.

| System Prompt | GPT-3's Response |
|---|---|
| You answer questions of a Broker exam. | Food and Drug Administration and Department of Health and Human Services. |
| You answer questions of a Broker exam. Answer with the explicit section used to answer the question. | Food and Drug Administration and Department of Health and Human Services. Section: 19 CFR Section (a) |
| Return the name of the section used to answer this question. | Animal and Plant Health Inspection Service and the Food and Drug Administration. Section used to answer the question: 19 CFR Section (a) Agencies within the Department of Homeland Security. |
| You locate sections of a legal document given a question. | The answer to the question cannot be found in the provided text. |

## 4 Experiments

The first section describes the different type of questions used to evaluate, the following two sections detail the adjustments made for each model, containing technical details of each implementation, as well as the corresponding workflow. Results are reported at the end of the section. Prior to the use of any model sections 111, 113, and the filtered questions were extracted from both the 19 CFR's XML version and the October Broker exam, and finally adjusted into a text file. No labeling was done since the original file contains a basic label for each section, subsection and sub-index. There was no preprocessing applied.

### 4.1 Questions

Broker exams work incredibly well for question answering systems since the U. S. Customs and Border Protection uses a guideline in order to asses the validity of each question presented [10]. Said guideline states that exam worthy questions should "be developed and answerable from a designated exam source" meaning the context provided in the books should be enough, and that they should "reflect real-world situations that a broker might encounter", favoring clear questions over ambiguous ones as well as those that require deduction only from the facts presented in the question.

These conditions generate questions without exaggerated vocabulary, that are in theory simple to answer, are complete with the facts required to answer, and most importantly vary in their levels of complexity. This last characteristic is vital since it deepens the model's need for reading comprehension capabilities. We will observe each models performance on different type of questions in the result section.

Knowing that RoBERTa and GPT-3 vary in a theoretical approach and technique, we need to evaluate them using a comparable procedure. In order to do that, both models will be asked each filtered exam question without any modification to them. The context input for both is the corresponding section required to answer the question.

**Table 3.** RoBERTa's answers.

| Q# | RoBERTa's answer | Exam answer |
|---|---|---|
| 66 | 5 years | 5 years |
| 69 | §\u2009111.96 | Respond to the summons by providing CBP with the entry files requested because the broker is legally obligated to make such records available for examination by CBP. |
| 75 | an individual, partnership, association, or corporate broker | An existing permit holder. |

### 4.2 RoBERTa

With access to HuggingFace's fine-tuned models specialized in Question Answering we avoided the process of fine-tuning and restrain our workflow to text preprocessing followed by the question based evaluation. We worked with a fine-tuned version of RoBERTa on the SQuAD2.0 [8] dataset. HuggingFace offers a simple procedure to perform Question Answering, we only need the chosen model, the sections that will serve as context, and the questions in a simple text form

Once readied the information with the use of a pipeline, we tokenized questions and context using the model's byte level BPE algorithm, we vectorized both elements and carried out the search in an efficient and simple way. RoBERTa answers with the most optimal response found in the text, meaning that even if the information retrieval procedure was correct we still need to check with the answer key since the answer might not be identical to the available ones in the multiple choice section, more about this results in 4.4.

### 4.3 GPT-3

In this case the workflow is more complex due to GPT-3 being able to perform tasks without the need of fine-tuning it first[1]. Nevertheless, we require additional procedure to be done so that the model is capable of answering based on information not presented during training. We worked with OpenAI's API and paid a fee to use GPT-3 unrestricted.

We followed two step workflow: the creation of embeddings for our datasets, and a search and evaluation over the embeddings using the corresponding questions. This strategy is optimal over fine tuning due to the fact that fine tuning adjusts the model's response to a specific type of prompt and differs from the Zero-shot learning approach. Similarly fine tuning doesn't alter the information the model has access to, making it unreliable for a QA task.

In order to create the embeddings, we used OpenAI's cl100k_base tokenizer without any preprocessing. Once tokenized, the embeddings are created using the text-embedding-ada-002 model that produces 1536-dimensional vectors. For further sections we chose the GPT-3 model gpt-3.5-turbo. GPT's responses are heavily reliant on the prompts given to the model. When working directly with the API, we have additional influence having access to a system prompt in addition to the user prompt.

**Table 4.** GPT-3's Answers.

| Q# | GPT-3's answer | Correct answer |
|---|---|---|
| 66 | The broker must retain the records for at least 5 years after the date of entry or final withdrawal for a warehouse entry. | 5 years. |
| 69 | The broker must comply with the Customs summons and provide CBP with the entry files for the ten entries prepared on behalf of Company Z within the last three years. | Respond to the summons by providing CBP with the entry files requested because the broker is legally obligated to make such records available for examination by CBP. |
| 75 | All licensed individuals, corporations, partnerships, and associations. | An existing permit holder. |

In table 2 we show an example of the varying answers depending on the system prompt used, the user prompt in this example is Q70: A broker filed an entry for an importer of Irish tea. In addition to retaining the Customs Documents required to make entry and file entry summary for tea, which partner government agencies' forms must also be filed and retained? The system prompt is depicted in the table.

Upon verifying these responses with the answer sheet, we observe that the first three prompts provide the correct answer, yet the second and third indicate an incorrect section in the exam. The fourth one gives an incorrect answer. We established our user prompts to be the Broker exam questions without any modifications, whereas the system prompts evaluated will be the following 2:

- **Prompt 1:** You answer questions of a broker exam.
- **Prompt 2:** You answer questions of a broker exam. Answer with the explicit section used to answer the question.

### 4.4 Results

Questions in Broker exams are not always answered by purely extracting text from the corresponding section. The referenced information and the logic behind the question are fundamental to answer adequately. The most important task in our experiments is the analysis of each model's response to these varying circumstances. The following questions taken out of the evaluation set serve as an example of this phenomenon.

- **Q66:** Generally, how many years after the date of entry or final withdrawal for a warehouse entry must the broker retain the records?
- **Q69:** A duly licensed customs broker was served and named in a Customs summons signed by the Director of the Consumer Products and Mass Merchandising Center. The summons requires the broker to provide CBP with the entry files for ten entries prepared on behalf of Company Z within the last three years. The broker terminated Company Z as a client one year ago.

**Table 5.** Exact matches.

| Model | Exact matches |
|---|---|
| RoBERTa | 0.31 |
| GPT-3 Prompt 1 | **0.59** |
| GPT-3 Prompt 2 | 0.5 |

**Table 6.** Exam performance.

| Model | Accuracy |
|---|---|
| RoBERTa | 0.19 |
| GPT-3 Prompt 1 | **0.69** |
| GPT-3 Prompt 2 | 0.5 |

Company Z has not provided the broker with any specific written instructions regarding responding to a summons within its now revoked power of attorney document. How must the broker respond to the Customs summons?

- **Q75:** Who listed below is responsible for paying the annual user fee detailed in Part 111 of the Customs Regulations? A) Importers who file their own entries. B) An existing permit holder. C) All licensed individuals, corporations, partnerships, and associations. D) A permit holder reporting monthly employee new hires and terminations. E) A licensed entity with an employee embedded at a client's facility.

We see that Q66 is similar to the vast majority of those in QA datasets asking a straight forward question without any additional context, Q69, on the other hand, poses a fictional scenario the reader must analyze in order to obtain the correct answer, finally as a middle ground Q75 tasks the reader to review possible answers before actually answering. For further reference we will divide the questions in two sections: traditional QA questions, reasoning based questions. Before moving onto the final evaluations, we expose in detail the way our models RoBERTa answers these questions in table 3.

Upon inspection, we first see that Q69's answer is far from a legible piece of text and that only Q66's answer matches the correct response. Nonetheless, if we search for the specific section of Q75's answer, we observe an extract from Section 111.96 (c). We realized that that is the correct section from which the answer is extracted, verified with the answer sheet. This suggests that even if RoBERTa answers poorly, the model's semantic search should, in theory, be capable to adequately match questions with sections.

On the other hand, GPT-3 presents a considerably improved answer given the model's text generation capabilities. Fiddling with different prompts we were able to obtain answers anywhere from single sentences to multiple paragraphs of length. The inconvenience presented by is that the answer does not always match the precise wording of the answers presented by the exam, even if the answer itself is fundamentally correct, as shown in table 4 with Q69. In a similar sense tracing the text used to generate the answer can not be done as directly as with RoBERTa, this can be seen in table 2 as GPT-3 returns a nonexisten section of the 19 CFR.

Higher complexity answers implie that we must evaluate these models using more than just the exact matches metric. We evaluated them using both the exact matches metric as well as the correct answer for the corresponding question determined by the writer's criteria. In order to determine each model's test performance, two writers independently reviewed the answer returned by the models and cross referenced them with the corresponding answer sheet. Tables 5 and 6 show the evaluation.

**Table 7.** Extract of the evaluation.

| Question | RoBERTa | GPT-3 (Prompt 1) |
|---|---|---|
| 3: What type of bond is needed to operate as a custodian of bonded merchandise? | Single transaction bond. | Basic Custodial Bond. |
| 73: Under what circumstances may a licensed broker accept fees from an attorney? | The charges are not paid bythe broker. | A licensed broker may accept fees from an attorney if the amount of the fee is commensurate with the time, effort, and skill expended by the broker in performing his services. |
| 76: What must a corporate broker do to continue to conduct Customs business after thecorporate office who qualified its license retires? | Provide written notice. | A corporate broker must provide written notice to the Assistant Commissioner when the corporate officer who qualified its license retires. The broker must also send a copy of the written notice to the director of each port through which a permit has been granted to the broker. |

To further discuss the results we present table 7 that shows 3 questions selected by their different complexity levels as well as each model's answers. The analysis of this table is done in the following section.

## 5 Analysis and Conclusion

RoBERTa is simple to implement and free option to work with. Nonetheless this model, mainly due to its encoder nature, lacks the capability to answer more complex questions as the ones shown in this paper. The questions used for evaluation ask more of the model rather than just locating the point in the text in which the answer is contained. Reasoning based on information is required in order to answer correctly questions such as Q69 and Q75, which tend to be the most frequent type of of question in the exam. GPT-3 as a decoder outperforms considerably its encoder counterpart, again, due to the nature of the exam questions.

This result suggests that language models that are able to analyze context and answer based on certain information are bound to perform better in complex scenarios of question answering. However GPT-3's prompt depending answers make it susceptible of a lower performance as seen in table 6. A more complex prompt, even if just by a sentence, endangers the model's performance on the test Table 7 exemplifies both model's behavior and limitations for this task.

For question 3 we see that both models manage to respond with a type of bond, even if RoBERTa's answer is incorrect (the answer is Bond Type 2) the model manages to return an clear answer to the question. GPT-3 answers correctly even though the exact match differs, a simple online search of what are Bond Type 2 returns Basic Custodial Bonds. Q73 provides an example of a question that RoBERTa can't answer correctly, its response is not related to the question. Remarkably GPT-3 answers almost exactly as the broker exam needs, this discrepancy might be due to the phrasing of the question itself.

Datasets like SQuAD contain mainly W and H questions[1] without modifiers e.g., What was the name of Beyoncé's second solo album? or What do greenhouses do with solar energy?. Even if broker exam questions are not ambiguous, their semantic properties make them more complex that the average QA question, hence decoder models answer them in a more efficient manner. Finally Q76 shows us that generative models are not optimal and that users should be critical with their answers.

GPT-3's response seems reasonable and well structured yet the correct answer differs substantially: Appoint a new broker as an officer of the corporation and notify CBP of the new license qualifier. Summarizing, decoder and generative models seem to be a better option for dealing with complex question answering scenarios. For this particular task even though RoBERTa's implementation was free and simple its performance overshadows these aspects. On the other hand even though GPT-3 might need a more complex workflow as well as a monetary involvement, its results are worth the effort. With minimal user prompt engineering (zero-shot scenario) the model achieved acceptable results.

## 6 Future Work

Future work requires a deeper and better structured dataset that contains all of the 19 CFR, not only specific sections, as well as the other four mentioned books that serve as a guide for Broker exams. Similarly a precise separation of subsections, as well as tagging might prove to be beneficial for section extraction with models like GPT-3. A long term project would be the creation of a dataset using complex questions specialized to extractive or generative models in order to carry out a fine tuning. Additionally, regarding the evaluation methodology, these experiments would benefit from verifying the answers with legal experts, brokers in particularly, in order to account for linguistic variations presented by generative models.

**Acknowledgments.** We thank CONAHCYT (CF-2023-G-64) and PAPIIT (IT100822) for the financial support for this paper.

## References

1. Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., Hesse, C., et al.: Language

[1] Who? What? When? Where? Why? How?

models are few-shot learners. In: Proceedings of the 34th International Conference on Neural Information Processing Systems (2020) doi: 10.48550/arXiv.2005.14165

2. Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., Grave, E., Ott, M., Zettlemoyer, L., Stoyanov, V.: Unsupervised cross-lingual representation learning at scale. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 8440–8451 (2020) doi: 10.18653/v1/2020.acl-main.747
3. Devlin, J., Chang, M. W., Lee, K., Toutanova, K.: BERT: Pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, vol. 1, pp. 4171–4186 (2019) doi: 10.48550/arXiv.1810.04805
4. Kaplan, J., McCandlish, S., Henighan, T., Brown, T. B., Chess, B., Child, R., Gray, S., Radford, A., Wu, J., Amodei, D.: Scaling Laws for Neural Language Models. arXiv (2020) doi: 10.48550/arXiv.2001.08361
5. Lewis, P., Oguz, B., Rinott, R., Riedel, S., Schwenk, H.: MLQA: Evaluating cross-lingual extractive question answering. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 7315–7330 (2019) doi: 10.18653/v1/2020.acl-main.653
6. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V.: RoBERTa: A robustly optimized BERT pretraining approach. In: Proceedings of the International Conference on Learning Representations, Conference Blind Submission, pp. 1–15 (2019) doi: 10.48550/arXiv.1907.11692
7. National Archives Records Administration's Office of the Federal Register and Government Publishing Office: Title 19 of the code of federal regulations. GovInfo (2022) www.govinfo.gov/app/collection/cfr/2022/title19/chapterI
8. Rajpurkar, P., Jia, R., Liang, P.: Know what you don't know: Unanswerable questions for SQuAD. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, vol. 2, pp. 784–789 (2018) doi: 10.18653/v1/p18-2124
9. Rajpurkar, P., Zhang, J., Lopyrev, K., Liang, P.: SQuAD: 100000+ questions for machine comprehension of text. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, pp. 2383–2392 (2016) doi: 10.18653/v1/d16-1264
10. U.S. Customs and Border Protection: Customs broker examination guidelines for writing new questions (2018) www.cbp.gov/trade/programs-administration/customs-brokers/license-examination-notice-examination/guidelines-writing-questions

ISSN 1870-4069

# Understanding and Tracking of Deformable Objects in the Context of Humanoid Robotics: A Literature Review

Gustavo De Los Ríos-Alatorre, Luis Alberto Muñoz-Ubando

Instituto Tecnológico y de
Estudios Superiores de Monterrey,
Mexico

gustavodlra1999@gmail.com, amunoz@tec.mx

**Abstract.** The following paper offers a comprehensive literature review of recent advancements and trends in the manipulation of deformable objects by robotics systems focusing as much as possible on the interaction between humanoid robotics and this type of objects. The paper boards different fields such as deep learning, traditional deformable object understanding, representations and handling, as well as the role that context awareness and multi-sensory data plays in the understanding of these objects. The explored works span a period around the last 30 years in order to get the most complete possible understanding.

**Keywords:** Computer vision, deep learning for visual perception, recognition.

## 1 Introduction

As a field of artificial intelligence, computer vision empowers computational systems to derive meaningful insights from digital images, videos, and other visual inputs, culminating in informed decision making [18]. This process frequently involves utilization of image processing techniques to aid in the interpretation of intricacies of a 3-D world from time varying 2-D data.

By allowing systems to perceive and understand their environment, computer vision paves the way for crucial applications in diverse sectors such as robotics, medicine, computer graphics and autonomous vehicles. Humanoid robotics consists of a field of study that aims to develop robots that mirror the structure and behavior of the human body, allowing them to function autonomously in environments designed for humans.

Fundamental concepts related to it include bipedal locomotion, balance control and complex motor functions that involve a blend of perception, decision-making, and execution. By integrating technologies from different domains such as mechanics, electronics and computer science, humanoid robots are designed to perform tasks such as walking, climbing stairs, or object manipulation, replicating human agility and dexterity.

Incorporating computer vision significantly enhances the capabilities of humanoid robots, enabling them to perceive and interact with their environment in a more refined way. It is particularly relevant to consider the tracking and manipulation of deformable or flexible objects, which represents an intricate challenge due to their geometries and material properties.

pp. 17–29; rec. 2023-06-04; acc. 2023-09-01

As the complexity of the geometry of an object and material properties increase, traditional computer vision techniques can often fail to deliver to the desired accuracy and robustness. The following text offers a comprehensive literature review focused on how modern computer vision strategies tackle these difficulties, exploring advancements in deep learning techniques that have enhanced tracking performance of flexible objects.

With an emphasis on the integration of object recognition and depth perception algorithms, this literature review intends on delving into the processes that enable precise, real-time tracking of these complex targets. The intersection of depth perception and object recognition forms a robust basis for the analysis of non-rigid objects, despite their inherent complexities.

Context-awareness also emerges as a significant aspect in the development of sophisticated computer vision algorithms. Incorporating contextual understanding allows humanoid robots to improve their interaction with deformable objects, facilitating more reliable tracking and manipulation.

Finally, this review presents an overview of how multi sensory data integration can augment the information gathered by computer vision techniques. Techniques to fuse data from different sensors are explored, such as tactile sensors, with visual data to enhance tracking and manipulation accuracy, particularly in complex environments.

Overall, this literature review aims to provide a thorough insight into the state-of-the-art in computer vision for flexible object tracking, focusing on the latest as well as the most significant advancements, challenges, and promising avenues for future research. The structure of the review will be based on the following research questions:

1. What unique challenges are presented by flexible object tracking for humanoids and how can computer vision effectively address these difficulties?
2. What are the advancements in deep learning techniques and how they can enhance the performance of vision-based tracking of flexible objects for humanoid robots?
3. How can object recognition and depth perception algorithms be integrated for precise real-time tracking of flexible objects?
4. What is the role of context-awareness in computer vision algorithms for improved tracking and manipulation of flexible objects by humanoid robots?
5. How can data from other sensors be integrated with computer vision to increase the accuracy of flexible object tracking and manipulation in complex environments?

## 2 Background and Historical Development

The first computer vision experiments begin in the 1960s with the experiments done by PhD. Lawrence Roberts, known as the father of computer vision. In his PhD Thesis for MIT many of the bases for computer vision techniques being used today were developed [36]. In this thesis, he managed to make a computer create a 3-D representation of solid objects in a photograph via the use of image transformations. The work done by Roberts establishes the foundations for future 3-D object recognition and tracking.

Throughout the 1980s and 1990s, deformable models were introduced. This advancement would prove fundamental for tracking deformable objects. Deformable models applicable to 2-D objects were introduced by Kass et al. as explicit deformable contours in 1988. In this early exploration, deformable contours were known as snakes, which are considered as an energy minimizing spline guided by external constraint forces and influenced by image forces that pull it towards features such as lines and edges [19].

Snakes are active contour models, locking into nearby edges and localizing them accurately. These deformable models were later generalized to work in 3-D by Terzuopolos et al. [43], their work on dynamic analytical models initiated the physically based deformable models. These models aimed to represent the physical properties and behavior of materials realistically, using principles from mechanics, physics, and biology.

In 1999, the Scale-Invariant Feature Transform (SIFT) algorithm was introduced by Lowe [28]. The main purpose of this algorithm is to extract information from features that are invariant to image scale and rotation, providing robust matching across a substantial range of distortion, noise or change in viewpoint. Applications of the SIFT algorithm include object tracking and recognition, 3-D reconstruction, and augmented reality.

Variations of the SIFT algorithm have also been utilized in the recognition of deformable objects. In [52], Zickler, S. et al. present an algorithm that utilizes PCA-SIFT in a combination with a clustered voting scheme. This model achieves detection and localization of multiple, highly deformable objects in real-time video footage.

Spanning the 2010s, the rise of deep learning methods saw a wide utilization of these methods in computer vison. Clear examples of these can be seen via the use of Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), particularly Long Short-Term Memory (LSTM) models being used for object recognition and tracking over time. This technology has been utilized in robotics-related tasks such as classification of slip occurrences by recognizing instances of objects being handled as well as their properties [24, 27, 51].

The main purpose behind this sort of application being object classification. Entering the 2020s, attention mechanisms and transformers which originated in the field of Natural Language Processing have been adapted for computer vision and robotics tasks. Interesting advancements have been made, such as in [40] were the authors proposed a Transformer framework for tabletop tasks encoding language goals and RGB-D voxel observations, outputting discretized 6-DoF actions.

The work done by Yunhai H. et al. [15] is also really promising regarding the utilization of deep learning-related models in tasks related to deformable object manipulation in robotics by utilizing a transformer-based robotic grasping framework for rigid robotic grippers that leverage tactile and visual information for safe grasping. The Transformers involved learn physical feature embeddings with sensor feedback predicting a grasping outcome via the use of a multilayer perceptron (MLP).

## 3 Flexible Object Tracking Challenges

The understanding and tracking deformable objects in real time applications would broaden the range of applications for robotics systems in areas such as surgery, household robotics, manufacturing, and those mentioned in previous sections.

A complete understanding of these objects is yet to be achieved, mainly due to their complex nature. Thus, representing, understanding, and manipulating deformable objects in robotics remains an open challenge, which has itself given rise to other significant challenges such as detecting the elasticity and plasticity of objects and manipulation planning and control.

Most of the work involving robotic manipulation tends to focus on working with rigid objects [7]. In [5] Billard et al. attribute this to the fact that flexible materials like fruits, vegetables and clothing have varied sizes, weights, and superficial properties, with manipulations that involve deformation being difficult due to the need for an accurate model that represents them. In most interactions, rigid objects are expected to keep their characteristics such as shape intact, with any forces applied to them represented as a series of rigid body transformations.

Conversely, deformable objects experience changes in their shape after a force is applied to them, with the degree of deformation and motion resulting from an interaction being dependent on the material composition of the object, making the physics of the deformation process hard to capture. Dynamics models have been utilized for high-fidelity mechanical modelling, however the information required to make these models is not always present for robotics systems in the real world [3].

All these factors combined with the fact that deformable objects can potentially have an infinite number of degrees of freedom due to their continuous nature [9], have resulted in current attempts of modelling, understanding, and tracking deformable objects being application specific [3]. On the vision front, the understanding of deformable objects is also made difficult because several elements in an image can increase the difficulty of detecting these objects throughout a sequence of frames.

Lighting conditions, background noise, and especially occlusion combined with the variety of appearances that a deformable object might have make it a significant challenge. Historically, these issues (especially occlusions) have been faced by complementing vision models with physics simulations, such as in [34] [42]. However, these approaches also suffer from needing a physical model which might not always be available. The uncertainty regarding the shape of the object also makes the task difficult since estimating all possible shape changes of an object is not easy with shape changes often being unpredictable.

Real-time processing of tasks involving deformable object tracking with computer vision remains a major challenge [25]. The complexity comes from the high dimensionality of deformable objects, needing to repetitively compute changes in the state of the object within a short time span. Obtaining results in a quick manner necessitates a trade-off between accuracy and speed. While training data exists for solutions that utilize artificial intelligence models [24] it is still relatively limited, mainly because, it is never possible to consider all deformations for the objects in a set [6].

This compounded by the fact that deformable objects are very diverse with individuals within groups of deformable objects presenting unique deformations. Learning models expected to work with deformable objects require a good capacity for generalization. Manipulation, planning, and control while handling deformable objects is also a significant challenge. Traditional planning and control techniques applied with rigid objects are not applicable [3].

The high dimensionality and non-linear nature of flexible objects requires the development of specialized case specific techniques, such as in [8, 47]. Significant strides continue to be made in the field; however, a long road still lies ahead. The challenges mentioned throughout this section are a reminder of the need for continuous research and development related to tracking and manipulation of these of objects. Upcoming sections will delve into some of the most promising advancements in the field.

## 4 Deep Learning Techniques for Object Tracking

The recent boom in machine learning that began in the latter part of the 2010s gave way to important developments in the realm of deep learning. Some of the most notable advancements include Convolutional Neural Networks [16, 13], Recurrent Neural Networks [32], Generative Adversarial Networks [10], Deep Reinforcement Learning [46] as well as Transformers and Attention Mechanisms [11].

Although these deep learning technologies were originally intended to be used with certain types of tasks in mind such as computer vision or natural language processing, the field of robotics has managed to adapt part of these technologies for tasks relevant to it. Convolutional Neural Networks have had a major role in expanding tracking capabilities in robotics systems.

For instance, in [29] the research team utilized a hierarchy of deep convolutional neural networks to categorize deformable objects as well as recognizing their pose. The framework used by the team utilizes two distinct CNN layers, the first one is used for classifying a certain garment based on predefined categories.

Meanwhile, the second layer consists of a category specific CNN to perform pose estimation, with the team testing their approach not only in simulations, but also with an actual robotic platform. In their implementation, the deformable garments are hung from a point to have a recognizable starting shape for the deformable object. The task of interacting with garments in a certain state is applicable in the industry while also being a desirable skill for future service robots.

The proposed pipeline achieved a recognition rate of 89.38 percent, outclassing state-of-the-art methods at the time of its release. Deep learning methods are making important contributions to the field of humanoid robotics. One notable study [50] outlines a novel approach for creating a humanoid robot capable of functioning in a manufacturing setting. This approach considers four principles for the robot and the algorithm: the ability to execute tasks, refine performance through repetition, adapt to new situations, and be easily implemented in real-world settings. Data for training is initially gathered through teleoperation.

The study employs a two-stage deep learning model, where a Deep Convolutional Autoencoder (DCAE) is responsible for image feature extraction and reconstruction, while a Time-Delay Neural Network (TDNN) learns the task dynamics based on these features and robot motion data. Since CNNs can handle considerably more input dimensions than fully connected neural networks with less parameters decreasing training time and enhancing the performance for image processing tasks such as feature or edge extraction on images, they present a solid alternative when working with vision tasks.

With a trained DCAE, half of the structure of the model is dedicated to encoding (compressing) the information to small-dimension image features which can represent the state of an input image and provide high-resolution input information in less dimensions, with batch normalization being used in the process to reduce the possibility of overfitting. The second half of the model is used to decode (reconstruct) the information to extracted image features.

The other neural architecture is the Time-Delay Neural Network (TDNN) [23]. TDNNs are designed to work with data that has a temporal component, making them ideal for tasks like sequence reconstruction. In [50], researchers employed a TDNN with multiple layers to generate continuous sequences by dynamically adjusting the input window over time.

This real-time adjustment is made possible by continuously feeding the network with new sensory and motor data, such as images captured by a camera and robot arm movements. The TDNN model in this study was capable of learning from multiple sensory-motor signal inputs, with the features extracted from the DCAE and robot motions being used by it. The experimental setup included a robot equipped with two 6-DOF non-back-drivable arms and a camera for precise manipulation.

Non-supervised learning has also been utilized to a great extent in relation to robotics and deformable object manipulation. For instance, in [31], the research team utilized a reinforcement learning approach to attempt to instruct a robot arm on how to interact with a deformable object in the form of cloth, considering actions such as diagonal folding, hanging, and folding the cloth up to a mark.

This was done utilizing a reinforcement learning technique known as Deep Deterministic Policy Gradients from Demonstrations (DDPGfD) [44], which is an extension of a prior algorithm known as Deep Deterministic Policy Gradients (DDPG) [26]. DDPG is a popular deep reinforcement learning technique used for solving continuous action space problems, meaning that the possible actions the agent can take exist in a continuous domain.

DDPGfD improves upon the original method by encouraging behavioral cloning, meaning that the agent that undergoes the reinforcement learning is rewarded by following a set of predetermined observations provided by an expert system. This helps mitigate usual reinforcement learning hassles such as slow convergence or having vast differences throughout iterations due to each of these iterations being conformed by random actions. The research team in charge of this study utilized the approach in both simulated and real-world environments, having a success rate of 90 percent when dealing with diagonal cloth folding in simulations while scoring a 66.6 percent grasp accuracy in the real world.

Transformer models carry with them great amounts of potential being capable of learning the data they are given in an intricate manner thanks to their attention mechanisms, which allow for models to focus on specific features of the input data that have a high relevance for the task at hand. In [15], the research team utilized the robotics specific transformer models TimeSformer [4] and ViViT [2] for the task of predicting slip detection when robotics systems handle fruits (deformable objects) which is relevant to factory and service robotics, outperforming models that made use of a CNN + LSTM pipeline that utilized Resnet18 [17].

The TimeSformer model works by processing spatial-temporal dimensions sequentially. The attention is applied first in the temporal dimension of the inputs at the same spatial position. Since these transformers work with computer vision tasks, the input image is dismantled into patches which are later flattened then linearly embedded to vectors of a certain size with positional embedding being added to each of them with a classifying token designed to extract task-level representations by tending to all other vectors. From there an input matrix is obtained and fed into a series of transformer layers.

The output of the classifying token is then used by a variety of tasks. The ViViT transformer works similarly to the TimeSformer model with a few key differences. For instance, ViViT processes the involved dimensions in parallel with half of the heads attending the spatial dimension and the other half the temporal dimension. The outputs are later combined via concatenation adding a linear transformation to half the size. It also has no need for a classification token, instead, the average of all output patches from the last Transformer layer is obtained passing it to a Multi-Layer Perceptron network (deep learning network used for classification) to predict a slip detection.

## 5 Integration of Object Recognition and Depth Perception Algorithms

Having a robotics system be capable of performing object recognition as well as having a certain ability to understand its environment via depth perception can be extremely beneficial towards working with deformable objects. Earlier research regarding the topic would utilize the extraction of visual features, such as silhouette features [22, 48, 21], however new more diverse approaches have been implemented partly thanks to the existence of low-cost RGB-D cameras.

These cameras provide both color (RGB) and depth information (D), enabling robots to better understand their environment in three dimensions. The two main currents regarding deformable object recognition tend to identify the object either when it is on a table [49, 41, 35] and those that recognize it when the object in question is hanging from a gripper [22, 12].

When the classification is done with the object on a table, it can be done with a single image, which is known as single shot perception. However, the scores obtained by single shot perception are usually outmatched by approaches where the robotics system interacts with the deformable object, gaining more information about the topology and deformations present in that object. In some instances, a 3-D model that comes because of the volumetric features in 3-D images.

**Table 1.** Data in the table is sourced from the original articles. Entries marked with ”NA” indicate that the information was not reported in the original study. The abbreviation ”FF” stands for Feed-Forward, used in the context of neural network-based methods.

| Category | Method | Year | Accuracy | Inference Time |
|---|---|---|---|---|
| Deep Learning | Inception-V3 CNN | 2018 | 88.03 % | Real-time |
| | TimeSformer | 2021 | 85.0 % | 2.46 s (FF) |
| | ViViT | 2021 | 83.9 % | 2.43 s (FF) |
| | Pose-recognition CNN | 2015 | 89.4 % | 1.8 s |
| | 2-phase DL Model | 2016 | 77.8 % | Real-time |
| | RL for manipulation | 2018 | 90 % (folding) | 24h |
| Deformation Models | Volumetric Approach | 2014 | 90 % (Shorts) | 0.22 s |
| | 3D DLO Shape detection | 2021 | 2.8 pixel error | 0.2 s |
| | Clothes state recognition | 2009 | 81.5 % | 5-20 s |
| | Model-driven clothes state estimation | 2002 | 72.41% | NA |

Considering a continuous perception of dynamic interactions focusing on the material and shape of a deformable object as it is being interacted with can be useful when trying to understand said object for classification purposes [30]. In [30] the research team adopts a framework inspired by [41] and [22].

The work in question presents and demonstrates a continuous visual perception approach for deformable object classification while a robot picks and observes how the object in question changes over time. The team extracts visual features from 2.5 D images in consecutive frames to learn a temporal-consistent representation of the dynamic attributes present in a particular piece of clothing.

First, the object is placed in a random configuration on a flat surface where a robot grasps it to observe its physical deformation. Different views of the object are utilized (egocentric and exocentric). This implementation was evaluated using two clothing databases with it working well for highly deformed garments. It obtained an accuracy of 66.7 percent among five categories, having an increase of 39.4 percent of classification score when considering other current approaches to clothing perception and recognition.

## 6 Context-Awareness and Multi Sensor Data Integration

Context-awareness in robotics and computer vision systems indicates that the systems that have this characteristic have a certain degree of knowledge about their environment. This knowledge is frequently obtained via a particular sensor such as a camera, a proximity sensor, an RGB-D sensor among others. Typical parts of a context may include the location, identity, activity and state of people, groups, and objects [38].

Physical variables can also be considered such as temperature and lighting. The field of collaborative robotics is in fact reliant in context-awareness. For operations that include collaborative robots and humans, context awareness must be timely in order to keep a safe and efficient working environment.

In [45], the research team suggests that visual observation of the motion of human workers may provide useful clues about tasks that need to be performed, with these having solid potential of being explored. Context awareness is also prevalent in autonomous vehicles, where it plays a central role in maintaining not only the safety of the autonomous vehicle and its occupants but also the safety of everything and everyone outside.

In [37] an advanced driving assistance system (ADAS) [1] is proposed and implemented with it being based on context-awareness of the environment that surrounds the vehicle. Ontological context awareness ADAS represent a significant advancement when compared to current systems, since these can use their context regardless of having to meet certain conditions, such as for the street that the driver is transiting to have lines.

Ontological ADAS also consider Mobile entities, static entities, and context parameters [1]. In many cases additional sensors or functionalities are suggested as would be the case with improving low-level object detection or better estimating road conditions [14], as well as more safety centered approaches [39]. A great way in which humanoid robots could acquire information about both their context and the state of the deformable object is via multisensory input.

Tactile sensor implementations alongside visual sensors are some of the most promising tactics regarding the understanding of deformable objects. For instance, in [15] visual data is combined with tactile data coming from a GelSight sensor to estimate when a slip may happen to a robot arm manipulating deformable objects such as fruits. Similar ideas such as visual servoing complemented with additional sensor data are explored in [20].

## 7 Conclusion

Advancements in the handling of deformable objects are being driven by a variety of evolving approaches. The integration of deep learning techniques, such as reinforcement learning, and the use of transformer models lay the foundation for even better models that may come while already delivering solid results. Also, the inclusion of multi-sensory inputs and context awareness will certainly improve the results that humanoid robots will have in terms of managing and interacting with deformable objects. There are still many challenges in the way but the stride towards the understanding of deformable objects is not stopping.

The tracking and understanding of deformable objects is particularly important in the development of robotics systems capable of working in unstructured industrial environments. Expanding the capacity of robots to understand deformable objects in this type of environments will allow for them to be utilized in more ample, novel applications, while working in environments with non-ideal conditions [33]. Some of the most significant models discussed in the review, based on criteria such as performance, novelty, and impact are present in Table 1.

**Acknowledgments.** Supported by ITESM.

## References

1. Armand, A., Filliat, D., Ibañez-Guzman, J.: Ontology-based context awareness for driving assistance systems. In: Proceedings of the IEEE Intelligent Vehicles Symposium Proceedings, pp. 227–233 (2014) doi: 10.1109/IVS.2014.6856509
2. Arnab, A., Dehghani, M., Heigold, G., Sun, C., Lučić, M., Schmid, C.: ViViT: A video vision transformer. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 6836–6846 (2021) doi: 10.1109/iccv48922.2021.00676
3. Arriola-Rios, V. E., Guler, P., Ficuciello, F., Kragic, D., Siciliano, B., Wyatt, J. L.: Modeling of deformable objects for robotic manipulation: A tutorial and review. Frontiers in Robotics and AI, vol. 7, pp. 82 (2020) doi: 10.3389/frobt.2020.00082
4. Bertasius, G., Wang, H., Torresani, L.: Is space-time attention all you need for video understanding? In: Proceedings of the 38th International Conference on Machine Learning, vol. 139, pp. 813–821 (2021) doi: 10.48550/ARXIV.2102.05095
5. Billard, A., Kragic, D.: Trends and challenges in robot manipulation. Science American Association for the Advancement of Science, vol. 364, no. 6446 (2019) doi: 10.1126/science.aat8414
6. Blanes, C., Mellado, M., Ortiz, C., Valera, A.: Review. Technologies for robot grippers in pick and place operations for fresh fruits and vegetables. Spanish Journal of Agricultural Research, vol. 9, no. 4, pp. 1130 (2011) doi: 10.5424/sjar/20110904-501-10
7. Bohg, J., Morales, A., Asfour, T., Kragic, D.: Data-driven grasp synthesis—A survey. IEEE Transactions on Robotics, vol. 30, no. 2, pp. 289–309 (2014) doi: 10.1109/tro.2013.2289018
8. Caporali, A., Galassi, K., Palli, G.: 3D DLO shape detection and grasp planning from multiple 2D views. In: Proceedings of the IEEE/ASME International Conference on Advanced Intelligent Mechatronics, pp. 424–429 (2021) doi: 10.1109/AIM46487.2021.9517655
9. Chi, C., Berenson, D.: Occlusion-robust deformable object tracking without physics simulation. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 6443–6450 (2019) doi: 10.1109/IROS40897.2019.8967827
10. Creswell, A., White, T., Dumoulin, V., Arulkumaran, K., Sengupta, B., Bharath, A. A.: Generative adversarial networks: An overview. IEEE Signal Processing Magazine, vol. 35, no. 1, pp. 53–65 (2018) doi: 10.1109/msp.2017.2765202
11. Devlin, J., Chang, M. W., Lee, K., Toutanova, K.: BERT: Pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, vol. 1, pp. 4171–4186 (2019)
12. Doumanoglou, A., Kim, T. K., Zhao, X., Malassiotis, S.: Active random forests: An application to autonomous unfolding of clothes. In: Proceedings of the European Conference on Computer Vision, pp. 644–658 (2014) doi: 10.1007/978-3-319-10602-1_42
13. Girshick, R.: Fast R-CNN. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1440–1448 (2015) doi: 10.1109/iccv.2015.169
14. Guo, C., Meguro, J., Kojima, Y., Naito, T.: A multimodal ADAS system for unmarked urban scenarios based on road context understanding. IEEE Transactions on Intelligent Transportation Systems, vol. 16, no. 4, pp. 1690–1704 (2015) doi: 10.1109/tits.2014.2368980
15. Han, Y., Batra, R., Boyd, N., Zhao, T., She, Y., Hutchinson, S., Zhao, Y.: Learning generalizable vision-tactile robotic grasping strategy for deformable objects via transformer. arXiv, pp. 1–11 (2021) doi: 10.48550/ARXIV.2112.06374
16. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask R-CNN. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2961–2969 (2017) doi: 10.48550/ARXIV.1703.06870

17. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016) doi: 10.1109/cvpr.2016.90
18. IBM: What is computer vision? (2023) www.ibm.com/topics/computer-vision
19. Kass, M., Witkin, A., Terzopoulos, D.: Snakes: Active contour models. International Journal of Computer Vision, vol. 1, no. 4, pp. 321–331 (1988) doi: 10.1007/bf00133570
20. Khalil, F. F., Payeur, P.: Dexterous robotic manipulation of deformable objects with multi-sensory feedback - A review. Robot Manipulators Trends and Development, pp. 587–619 (2010) doi: 10.5772/9183
21. Kita, Y., Kita, N.: A model-driven method of estimating the state of clothes for manipulating it. In: Proceedings of the 6th IEEE Workshop on Applications of Computer Vision, pp. 63–69 (2002) doi: 10.1109/acv.2002.1182158
22. Kita, Y., Ueshiba, T., Kanehiro, F., Kita, N.: Recognizing clothing states using 3D data observed from multiple directions. In: Proceedings of the 13th IEEE-RAS International Conference on Humanoid Robots, pp. 227–233 (2013) doi: 10.1109/humanoids.2013.7029980
23. Lang, K. J., Waibel, A. H., Hinton, G. E.: A time-delay neural network architecture for isolated word recognition. Neural Networks, vol. 3, no. 1, pp. 23–43 (1990) doi: 10.1016/0893-6080(90)90044-1
24. Li, J., Dong, S., Adelson, E.: Slip detection with combined tactile and visual information. In: Proceedings of the International Conference on Robotics and Automation, pp. 7772–7777 doi: 10.48550/ARXIV.1802.10153
25. Li, Y., Wang, Y., Case, M., Chang, S. F., Allen, P. K.: Real-time pose estimation of deformable objects using a volumetric approach. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 1046–1052 (2014) doi: 10.1109/iros.2014.6942687
26. Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., Wierstra, D.: Continuous control with deep reinforcement learning. In: Proceedings of the International Conference on Learning Representations, pp. 1–14 (2016)
27. Lin, J., Calandra, R., Levine, S.: Learning to identify object instances by touch: Tactile recognition via multimodal matching. In: Proceedings of the International Conference on Robotics and Automation, pp. 3644–3650 (2019) doi: 10.1109/ICRA.2019.8793885
28. Lowe, D. G.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision, vol. 60, no. 2, pp. 91–110 (2004) doi: 10.1023/b:visi.0000029664.99615.94
29. Mariolis, I., Peleka, G., Kargakos, A., Malassiotis, S.: Pose and category recognition of highly deformable objects using deep learning. In: Proceedings of the International Conference on Advanced Robotics, pp. 655–662 (2015) doi: 10.1109/icar.2015.7251526
30. Martínez, L., Ruiz-del Solar, J., Sun, L., Siebert, J. P., Aragon-Camarasa, G.: Continuous perception for deformable objects understanding. Robotics and Autonomous Systems, vol. 118, pp. 220–230 (2019) doi: 10.1016/j.robot.2019.05.010
31. Matas, J., James, S., Davison, A. J.: Sim-to-real reinforcement learning for deformable object manipulation. In: Proceedings of The 2nd Conference on Robot Learning. Proceedings of Machine Learning Research, pp. 734–743 (2018)
32. Medsker, L. R., Jain, L.: Recurrent neural networks: Design and Applications. Chemical Rubber Co Press, Inc (1999)
33. Mitrano, P., MConachie, D., Berenson, D.: Learning where to trust unreliable models in an unstructured world for deformable object manipulation. Science Robotics, vol. 6, no. 54 (2021) doi: 10.1126/scirobotics.abd8170

34. Navarro-Alarcon, D., Yip, H. M., Wang, Z., Liu, Y. H., Zhong, F., Zhang, T., Li, P.: Automatic 3-D manipulation of soft objects by robotic arms with an adaptive deformation model. IEEE Transactions on Robotics, vol. 32, no. 2, pp. 429–441 (2016) doi: 10.1109/tro.2016.2533639
35. Ramisa, A., Alenya, G., Moreno-Noguer, F., Torras, C.: Using depth and appearance features for informed robot grasping of highly wrinkled clothes. In: Proceedings of the IEEE International Conference on Robotics and Automation, pp. 1703–1708 (2012) doi: 10.1109/ICRA.2012.6225045
36. Roberts, L.: Machine perception of three-dimensional solids. Massachusetts Institute of Technology (1963)
37. Ryu, M., Cha, S. H.: Context-awareness based driving assistance system for autonomous vehicles. International Journal of Control and Automation, vol. 11, no. 1, pp. 153–162 (2018) doi: 10.14257/ijca.2018.11.1.14
38. Salber, D., Dey, A. K., Abowd, G. D.: The context toolkit: Aiding the development of context-enabled applications. In: Proceedings of the Special Interest Group on Computer-Human Interaction Conference on Human Factors in Computing Systems, pp. 434–441 (1999) doi: 10.1145/302979.303126
39. Shen, Y., Jeong, J., Oh, T., Son, S. H.: CASD: A framework of context-awareness safety driving in vehicular networks. In: Proceedings of the 30th International Conference on Advanced Information Networking and Applications Workshops, pp. 252–257 (2016) doi: 10.1109/WAINA.2016.74
40. Shridhar, M., Manuelli, L., Fox, D.: Perceiver-actor: A multi-task transformer for robotic manipulation. In: Proceedings of the 6th Conference on Robot Learning, pp. 1–28 (2022) doi: 10.48550/ARXIV.2209.05451
41. Sun, L., Aragon-Camarasa, G., Rogers, S., Stolkin, R., Siebert, J. P.: Single-shot clothing category recognition in free-configurations with application to autonomous clothes sorting. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 6699–6706 (2017) doi: 10.1109/IROS.2017.8206586
42. Tang, T., Fan, Y., Lin, H. C., Tomizuka, M.: State estimation for deformable objects by point registration and dynamic simulation. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 2427–2433 (2017) doi: 10.1109/IROS.2017.8206058
43. Terzopoulos, D., Witkin, A., Kass, M.: Constraints on deformable models: Recovering 3D shape and nonrigid motion. Artificial Intelligence, vol. 36, no. 1, pp. 91–123 (1988) doi: 10.1016/0004-3702(88)90080-x
44. Vecerik, M., Hester, T., Scholz, J., Wang, F., Pietquin, O., Piot, B., Heess, N., Rothörl, T., Lampe, T., Riedmiller, M.: Leveraging demonstrations for deep reinforcement learning on robotics problems with sparse rewards. arXiv, pp. 1–10 (2017) doi: 10.48550/ARXIV.1707.08817
45. Wang, P., Liu, H., Wang, L., Gao, R. X.: Deep learning-based human motion recognition for predictive context-aware human-robot collaboration. CIRP Annals, vol. 67, no. 1, pp. 17–20 (2018) doi: 10.1016/j.cirp.2018.04.066
46. Watkins, C. J. C. H., Dayan, P.: Q-learning. Machine Learning, vol. 8, no. 3–4, pp. 279–292 (1992) doi: 10.1007/bf00992698
47. Wi, Y., Florence, P., Zeng, A., Fazeli, N.: VIRDO: Visio-tactile implicit representations of deformable objects. In: Proceedings of the International Conference on Robotics and Automation, pp. 3583–3590 (2022) doi: 10.1109/icra46639.2022.9812097
48. Willimon, B., Birchfield, S., Walker, I.: Model for unfolding laundry using interactive perception. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 4871–4876 (2011) doi: 10.1109/IROS.2011.6095066
49. Willimon, B., Walker, I., Birchfield, S.: Classification of clothing using midlevel layers. ISRN Robotics, vol. 2013, pp. 1–17 (2013) doi: 10.5402/2013/630579

50. Yang, P. C., Sasaki, K., Suzuki, K., Kase, K., Sugano, S., Ogata, T.: Repeatable folding task by humanoid robot worker using deep learning. IEEE Robotics and Automation Letters, vol. 2, no. 2, pp. 397–403 (2017) doi: 10.1109/lra.2016.2633383
51. Yuan, W., Wang, S., Dong, S., Adelson, E.: Connecting look and feel: Associating the visual and tactile properties of physical materials. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4494–4502 (2017) doi: 10.1109/cvpr.2017.478
52. Zickler, S., Efros, A.: Detection of multiple deformable objects using PCA-SIFT. In: Proceedings of the 22nd National Conference on Artificial Intelligence, vol. 2, pp. 1127–1132 (2007)

ISSN 1870-4069

# Relational Representation for Knowledge Transfer in Reinforcement Learning

Armando Martínez, Eduardo F. Morales,
L. Enrique Sucar

Instituto Nacional de Astrofísica, Óptica y Electrónica,
Coordinación de Ciencias Computacionales,
Mexico

{armandomtzruiz, emorales, esucar}@inaoep.mx

**Abstract.** In reinforcement learning, an agent learns to behave, through trial and error, in a dynamic environment. This way of learning requires a considerable amount of data and time. Relational representation allows to abstract the state space, and therefore it can help the learning process to be faster, and also, abstract representations are easier to transfer to other similar domains that share these abstractions. In this paper we investigate whether the agent learns faster using these methods, as well as verify that this abstraction is transferable to other tasks. The proposed approach was evaluated on learning to play different ATARI games with promising results.

**Keywords:** Reinforcement learning, relational representation, causal models, transfer learning, object detection.

## 1 Introduction

Reinforcement learning is a paradigm of machine learning that studies how an agent maximizes a future reward it receives from the environment through its interactions with the environment [10]. At each iteration, the agent receives a signal from its current state $s$ and selects an action $a$. The action possibly changes the state and the agent receives a reward signal $r$. The goal is to find a policy, which is a function that maps states to actions, that maximizes the total expected reward.

One of the problems with reinforcement learning is that the interactions of an agent and the environment in which it finds itself require a considerable amount of data and time for the agent to learn, so the learning process in this approach is time consuming. In addition, reinforcement learning suffers from its poor ability to generalize learned knowledge to new, although related problems.

Based on the above, the use of relational representation can allow us to abstract certain characteristics of the environment, so that the agent's learning can be accelerated [7]; also, this more abstract representation facilitates policy transfer to related tasks. In the experiments conducted in this work, the Arcade Learning Environment (ALE) library in Python was used to model Atari games as reinforcement learning environments. Relationships representing states within the environment were defined and the Q-Learning algorithm was adapted to incorporate relational representations into the agent's decision-making process.

pp. 31–44; rec. 2023-06-04; acc. 2023-09-01

**Table 1.** Summary of the most relevant related work.

| Paper | Reference | Year | Approach | Key Contributions |
|---|---|---|---|---|
| Deep-Q Networks | [6] | 2015 | Deep Q-Learning | Introduced the Deep Q-Network (DQN) for Atari games, pioneering deep RL by achieving human-level performance. Leveraged experince replay and target networks for stability. |
| Continuous control with deep RL | [4] | 2019 | Deep Q-Learning | Introduces a deep RL algorithm for continuous control, leveraging an actor critic method with neural function approximators. |
| Transfer Learning in RL | [1] | 2020 | Knowledge Graphs, Transfer Learning | Proposed knowledge graph-based transfer learning, outperforming existing methods. |
| Towards deep symbolic RL | [3] | 2016 | Deep Learning, Symbolic Reasoning | Presented a hybrid approach fusing deep neural networks with symbolic reasoning for complex tasks, surpassing pure deep and symbolic methods in a 3D maze navigation task. |
| Deep Hierarchical Approach to Lifelong Learning in Minecraft | [11] | 2016 | Deep Hierarchical RL, Minecraft | Introduced hierarchical RL approach for lifelong learning in Minecraft with the Deep Skill Module, enabling high-level skill acquisition and reuse. Demonstrated adaptability to various tasks in Minecraft, with implications for complex, open-ended environments beyond gaming. |

Although several relations can become true at certain stage, only one is considered, reducing this way the state-action space required.The experimental results obtained in learning to play several Atari games showed that the proposed approach achieved results comparable to existing works (e.g., DQN) in the selected games.

In particular, an improvement in learning speed and knowledge transfer between similar games was observed. Futhermore, it was shown that the incorporation of relational representations in the agent's decision making process allowed a better understanding of the environment and effective decision making.

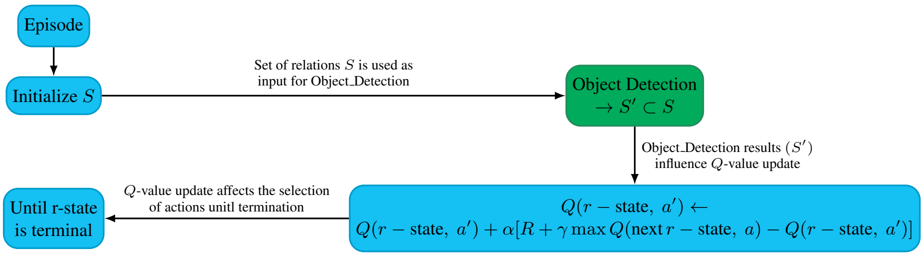

Episode
Initialize $S$
Set of relations $S$ is used as input for Object_Detection
Object Detection $\rightarrow S' \subset S$
Object_Detection results $(S')$ influence $Q$-value update
$Q(r-\text{state},\ a') \leftarrow Q(r-\text{state},\ a') + \alpha[R + \gamma \max Q(\text{next}\, r-\text{state},\ a) - Q(r-\text{state},\ a')]$
$Q$-value update affects the selection of actions unitl termination
Until r-state is terminal

**Fig. 1.** Block diagram of the Q-Learning algorithm with the incorporation of the Object Detection system. The Object Detection system is used to detect and classify the agent and the objects in its environment (trophies, obstacles, and walls). By incorporating the Object Detection system into the Q-Learning algorithm, the agent can learn to associate its actions with the objects in its environment and learn an optimal policy more efficiently.

## 2 Background

### 2.1 Markov Decision Processes

A Markov decision process (MDP) models a sequential decision problem, where an agent controls a system in a dynamic environment. The solution of an MDP consists of a sequence of states that maximize the expected reward, called a policy. Formally, a Markov decision process is a stochastic process defined as follows: Let $S = \{s_1, s_2, \cdots, s_n\}$ and $A = \{a_1, a_2, \cdots, a_m\}$ be finite sets with $n, m > 0$, where $S$ represents the set of states where the agent can be found and $A$ represents the set of actions the agent can perform. Then, a Markov decision process is a tuple $M =< S, A, \phi, r >$, where $\phi$ is defined as:

$$\phi : A \times S \times S \longrightarrow [1, 0]. \tag{1}$$

Which represents the transition of the specified agent as a probability distribution, and $R$ defined as:

$$r : S \times A \longrightarrow \mathbb{R}. \tag{2}$$

Is the reward function. A policy $\pi$ is a function $\pi : S \longrightarrow A$ that determines what action $a \in A$ to perform given a state $s \in S$. We define an optimal action-value function $Q^*(s, a)$ as the maximum expected return value:

$$Q^*(s, a) = \max_{\pi} \mathbb{E}[R_t \mid s_t = s,\ a_t = a,\ \pi], \tag{3}$$

where $R = \sum_{t'=t}^{T} \gamma^{t'-t} r_{t'}$ represents the future discounted return at time $t$.

## 3 Related Work

In this section, we briefly review previous work on deep RL, transfer learning and relational representations for RL.

**Algorithm 1** RQ-Learning.

```
Algorithm parameters: $\alpha \in [0,1]$, $\gamma \in [0,1]$, $\epsilon > 0$ small (between 0.1 and 0.01).
Initialize $Q(s,\ a), \forall s \in S,\ a \in A$ in arbitrary way.
for each episode do
    Initialize $S$ (set of relations)
    while $r$-state is not terminal do
        $S' = \text{Object\_Detection(frame)} \subset S$
        $r - \text{state} = \text{best\_relation}(S')$
        Choose $a'$ from $s$ using policy derived from $Q$ ($\epsilon$ greedy)
        Observe $R(r - \text{state},\ a') = R$
        $S'' \leftarrow \text{Object\_Detection(next frame)} \subset S$
        $\text{next\_r} - \text{state} = \text{best\_relation}(S'')$
        $Q(r-\text{state},\ a') \leftarrow Q(r-\text{state},\ a') + \alpha[R + \gamma \max_a Q(\text{next\_}r-\text{state},\ a) - Q(r-\text{state},\ a')]$
        $r - \text{state} \leftarrow \text{next\_r} - \text{state}$
    end while
end for
```

### 3.1 Deep Q-Networks

The fundamental work by Mnih et al. (2015) titled “Human-level control through deep reinforcement learning” [6] has made significant contributions to the field of deep reinforcement learning (DRL). The authors introduced the Deep Q-Network (DQN) algorithm, which is a powerful deep reinforcement learning algorithm capable of learning to play Atari games at or exceeding human-level performance. DQN is a deep neural network that predicts the expected reward for an action in a given state. This algorithm uses a strategy called experience replay, which consists in storing the agent’s experiences in a replay memory and randomly samples from it during training.

An important work related to DQN is the work of Lillicrap et al. [4], where the focus is a DQN algorithm that handles a continuous action space, which uses an iterative optimization process to find the action that maximizes the action-value function in continuous-valued environments. In contrast with the work by Mnih, the approach introduced here employs an actor-critic method with neural function approximators to directly learn a policy from raw sensory input, enabling it to manage continuous and high-dimensional action spaces.

### 3.2 Transfer Learning in Reinforcement Learning

Rajeswaran et al. (2020) introduced a new way to improve deep reinforcement learning using prior knowledge defined as knowledge graphs [1], which are structured representations of information about a domain. These graphs include entities, relationships between entities, and attributes for each entity. This approach involves training a knowledge graph using data from a source task and then transferring this knowledge to a target task. They use the Deep Q-Network algorithm that incorporates the knowledge graph into the process of decision making by the agent. In their evaluation on different transfer learning tasks, they found that their approach performs better than existing methods.

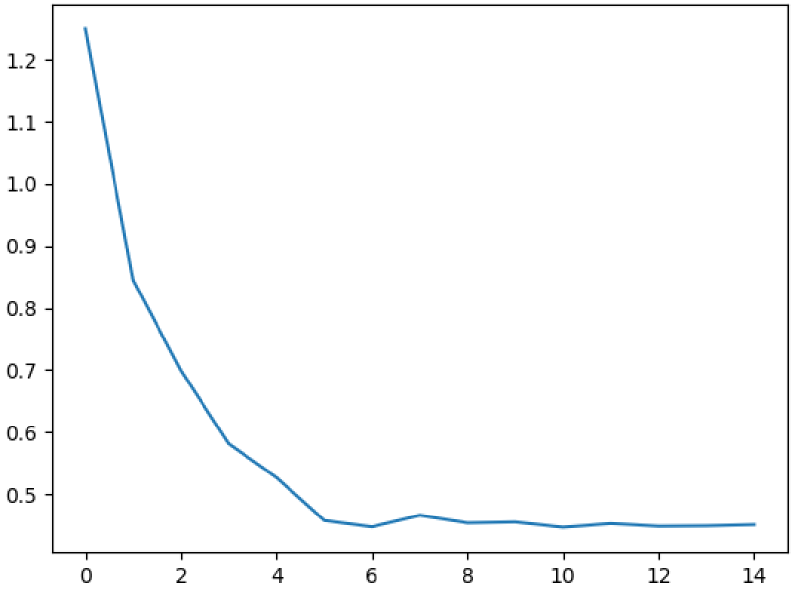


**Fig. 2.** Loss function.

They also show that the knowledge graph can be updated during learning, allowing the agent to adapt to changes in the environment. Tessler et al.'s paper, "A Deep Hierarchical Approach to Lifelong Learning in Minecraft," introduces a novel approach to reinforcement learning within the environment of the game Minecraft. The authors tackle the challenge of enabling an agent to acquire a diverse range of skills and adapt continuously to the dynamic Minecraft enfironment. They propose a hierarchical architecture, which uses the Deep Skill Module.

This module empowers the agent to learn, reuse, and combine high-level skills, which are generated separately by training the agent in specific tasks. The approach is grounded in a combination of reinforcement learning techniques and deep neural networks, which collectively provide a powerful framework for tackling the challenges posed by tasks in complex videogame environments, such as Minecraft.

### 3.3 Relational Representation in Reinforcement Learning

Garnelo et al. (2016) proposed an innovative way to tackle complex tasks in reinforcement learning [3]. They acknowledge that deep learning is powerful when it comes to learning from experience, meaning it can excel at learning patterns and representations from large amounts of data. It's particularly effective when dealing with high-dimensional inputs, such as images or raw sensor data.

However, there are certain tasks in reinforcement learning that require what they call "symbolic reasoning". Symbolic reasoning involves using logical rules and relations to understand abstract concepts and symbols. To overcome this, they present a hybrid approach that combines deep learning with a symbolic reasoning engine. This approach can be divided in two parts:

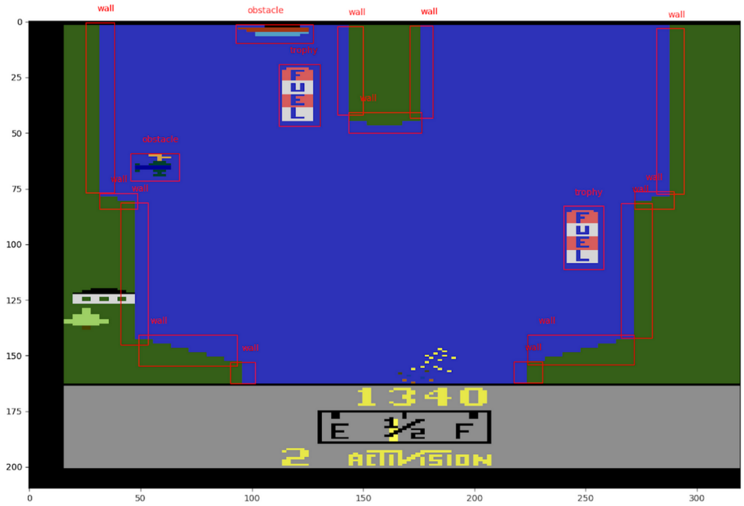


**Fig. 3.** Real-time object detection in a video game environment.

1. Deep learning component: This part uses a deep neural network to learn a policy from experience, where it was used a variant of the Deep Q-Network algorithm.
2. Symbolic reasoning engine: This component handles the symbolic reasoning of the agent within the environment. A first order logic engine is used to represent the actions performed by the agent in a symbolic manner.

To test their approach, the authors applied it to a 3D maze navigation task. They found that their approach outperformed both a pure deep learning approach and a pure symbolic reasoning approach. Table 1 summarizes the studies related to this work. In contrast to prior work, our approach directly employs images to extract object relations within the environment. This method may enhance the agent's learning efficiency, facilitating the transfer of the earned policy to similar games.

## 4 Proposed Approach

The block diagram in Figure 1 is a visual representation of the proposed approach. It is based on two main elements: (i) an object detection system that identifies and classifies the main visual elements in the game environment; and (ii) a relational representation of the states based on the detected objects and their spatial relations with the agent. Next we describe both elements.

### 4.1 State and Action Representation

A relational representation is a way of representing a state in the environment or knowledge about it in terms of entities and their relationships. This representation allows to model in a more expressive way the domains in such a way that some aspects of the environment can be generalized.

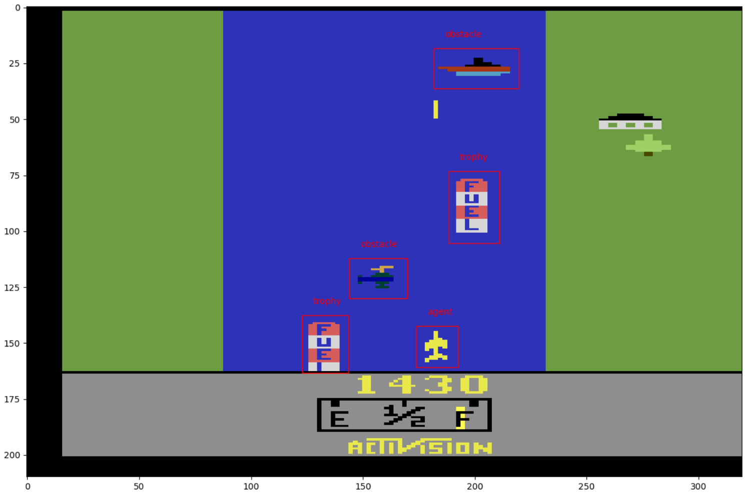


**Fig. 4.** Example of relational representation through object detection.

In this case, we used spatial relations to make an abstraction of the environment the agent is in. These relations where defined according to the games selected for the experiments, which are the following:

$$\text{CloseToObject(Agent, Object)},\\ \text{FarFromObject(Agent, Object)},\\ \text{CloseToWall(Agent, Wall)},$$

where Object can take the values obstacle or trophy, and Wall can take the values $R$, $L$, $UR$, $UL$, $F$, $B$ to denote the right sidewall, left sidewall, right top wall, left top wall, top wall, and bottom wall respectively. Since first order logic allows to give truth value to these relations, one can represent the states of the environment with disjunctions between the relations, i.e., the states within the environment have the following possible representation:

$$\text{CloseToObject(Agent, Object)} \vee \text{CloseToWall(Agent, Object)},\\ \text{FarFromObject(Agent, Object)} \vee \text{CloseToWall(Agent, Object)}.$$

Then, the set of states $S$ was defined with 119 relations that allow to make an abstraction of the elements surrounding the agent. These relations were defined by considering possible combinations of variables (obstacle, trophy and wall), focusing solely on detectable combinations. Specifically, we considered only those relationships that can occur within the object detection process during the agent's trainnig.

In this paper, the states defined by relations are called $r-$states. The proposed approach is applied to ATARI 2600 game environments. The choice of this type of environments is due to the fact that important features can be generalized using relationships between the agent and its environment. The ALE library for Python allows to define the set of actions as integers, i.e., the set of actions is predefined according to the possible movements of the Atari control.

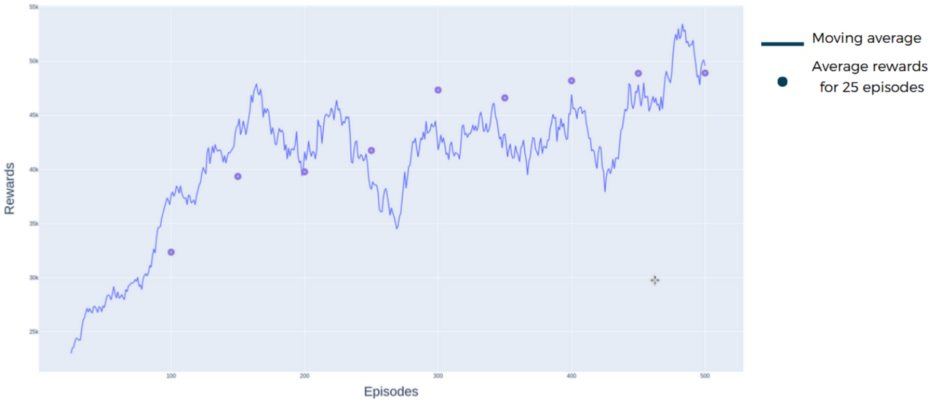


**Fig. 5.** Episodes vs Rewards. This graph represents the average rewards every 50 episodes to observe the increase in rewards as the episodes progress.

Since these 18 control actions (NOOP, up, down, left, right, upleft, upright, downleft, downright, fire, upfire, downfire, leftfire, rightfire, upleftfire, uprightfire, downleftfire and downrightfire) can be represented by integers in the order in which they were listed, the set of actions is then defined as follows:

$$\mathbb{A} = \{0,\ 1,\ 2,\ \cdots,\ 15,\ 16,\ 17\}. \tag{4}$$

Having the set of states represented as a set of relations and the set of actions defined as a set of integers representing each possible action that can be executed with the Atari controller, the reward function was defined as follows, given $k, m, n \in \mathbb{R}$:

$$R(r - \text{state},\ \text{action}) = \begin{cases} k & \text{if the agent is near a trophy,} \\ m & \text{if the agent is near an obstacle,} \\ n & \text{if the agent is near a wall.} \end{cases} \tag{5}$$

In this work, the decision was made to assign a higher reward to the agent for collecting trophies compared to the reward for moving away from an obstacle. In addition, it was defined that the highest reward the agent can receive is for moving away from the walls. To achieve this, the following value relation was established: $m < k < n$.

### 4.2 Object Detection

The incorporation of these elements to the Q-Learning algorithm is based on object detection, since through it the current state of the agent can be defined. The object detection system is implemented using the Python Detecto library, based on the Faster R-CNN architecture. This architecture consists of two modules: a deep convolutional network for region processing (the Region Proposal Network or RPN) and a Fast R-CNN detector using theses proposed regions.

**Table 2.** Comparison of average game scores between DQN and RQ-Learning approaches in Atari game environments.

| Game | DQN | Frames | Q-Learning + OBD | Frames |
|---|---|---|---|---|
| River Raid | 3166.2 (125.2) | 10M | 2667.1 (761.7) | 8.4M |

These regions define environmental objects, enabling us to identify spatial relations between the agent and its surroundings. Moreover, since object detection results in a subset $S'$ of the set of states $S$, it is necessary to define a state selection criterion. In each state there can be relations with several objects. As the number of objects increases, the possible number of states grows exponentially (combinations of all the relationships of all the objects).

One way to avoid this and simplify the problem is to look only at the relationship with the object that yields the best value in the Q function. For this purpose, it was decided by choosing the relation or $r-$state with the largest Q-value, and in case of ties, a relation is chosen at random. Formally, the choice of the $r-$state can be represented by the following function:

$$r - state = \begin{cases} R_i & \text{if } \arg\max Q(S', a) = R_i,\ \forall a \in \mathbb{A},\ i = 1,\, 2,\, \cdots,\, |S'|, \\ \mathcal{R} & \text{otherwise,} \end{cases} \tag{6}$$

where $\mathcal{R}$ is a random variable with probability function:

$$P(\mathcal{R} = R_i) = \frac{1}{|S|},\ i = 1,\, 2,\, \cdots,\, |S'|. \tag{7}$$

Therefore, the proposed Q-Learning algorithm, which includes the incorporation of the object detection system is described in Algorithm 1. Notice that the function best_relation is equivalent to the defined selection of $r$-states based on the subset of relations found by the object detection system.

It is important to mention that the object detection is based on the images extracted in real time from the game during the episode. In addition, in order to compare the results obtained in this work with other similar ones, we chose to process every 4 frames the image of the game to extract the relationships of the agent with the environment.

## 5 Experimental Results

### 5.1 Arcade Learning Environment

For this work, the Arcade Learning Environment (ALE) library for Python was used to model Atari games as reinforcement learning environments. In particular, games were selected from the shoot'em up category, where the main objective is to survive waves of enemy attacks while trying to destroy them or evade obstacles and enemies. These games provide fast-paced gameplay challenges, with intense action and levels that increase in difficulty as the player progresses through the game. Video games serve as ideal environments for reinforcement learning for several reasons.

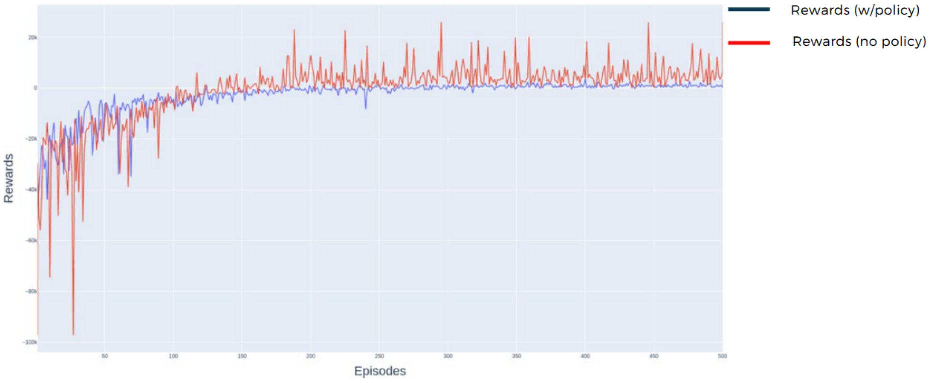


**Fig. 6.** Comparison of the performance of the agent in the game Chopper Command with a previously learned policy. The x-axis of the graph represents the number of episodes, while the y-axis represents the average reward obtained by the agent over the last 1000 episodes. The red line represents the performance of the agent when learning from scratch, while the blue line represents the performance of the agent when using a policy learned in a different game (River Raid) as a starting point.

Firstly, they offer well-defined settings with clear objectives and specific actions, enabling the direct observation and precise measurement of an agent's actions and outcomes. Secondly, games can simulate intricate scenarios, often impossible to replicate in reality, facilitating the exploration of reinforcement learning aspects such as decision-making, strategic planning, and adaptation to dynamic environments.

Thirdly, their scalability allows to assess an agent's ability to learn and adapt to increasingly complex tasks. Additionally, games generate substantial data, including state information, actions, and rewards, which prove invaluable for training and evaluating reinforcement learning algorithms. Lastly, the repetitive nature of games enables consistent task repetition, simplifying experimental replication and result comparisons among different approaches, making them a favored domain for machine learning research.

### 5.2 Object Detection

The object detection model was trained using a dataset comprising 65 images, aiming to accurately detect and classify both the agent and objects within its environment, including trophies, obstacles, and walls. Figures 2 and 3 illustrate the loss function graph for the detection model and provide an example of object detection within an ATARI 2600 game environment.

This detection process is conducted periodically during the agent's training via the ALE library. It involves extracting the image at a specific time $t$ and subsequently processing it through the trained object detection system model. The model generates a set of tensors that represent the coordinates of the bounding boxes' extreme points for objects detected within the given frame.

The relations between the agent and surrounding objects are defined based on object detection. Based on Figure 4, one can define the relation of the agent and the nearest object (obstacle) using first-order logic [9] as follows:

CloseToObject$(a,\ o)$, which is true, if agent $a$ is near object $o$,
CloseToWall$(a,\ o)$, which is true, if agent $a$ is near wall $w$,
FarFromObject$(a,\ o)$, which is true, if agent $a$ is far from object $o$.

For the objects, the relations were defined using the euclidean distance between the centroid of the agent and the centroid of the object. Let $m,\ n \in \mathbb{R}$ different than 0 with $m < n$, if $(x_a,\ y_a)$ represents the centroid of the agent and $(x_o,\ y_o)$ represents the centroid of the obstacle, then the relation CloseToObject$(a,\ o)$ is true if:

$$d((x_a,\ y_a),\ (x_o,\ y_o)) = \sqrt{(x_a - x_o)^2 + (y_a - y_o)^2} < m. \tag{8}$$

And the relation FarFromObject$(a, o)$ is true if:

$$m < d((x_a, y_a), (x_o, y_o)) < n. \tag{9}$$

In the case of walls, the CloseToWall relation was defined using the standard Euclidean distance between two real numbers, along with constraints imposed on the position of the agent with respect to the position of the wall. So the first-order logic expression of the relation is:

$$\begin{gathered}\exists a[\text{Rel}(a,\ \text{obstacle},\ \text{left},\ \text{up}, \emptyset) \iff \\ \text{CloseToObject}(a,\ \text{obstacle}) \vee \text{CloseToWall}(a,\ w) \wedge x = \text{left} \wedge y = \text{up} \wedge w = \emptyset], \\ \text{obstacle} \in \hat{O},\ x \in \hat{X},\ y \in \hat{Y},\ w \in \hat{W},\end{gathered} \tag{10}$$

where $\hat{O} = \{\emptyset,\ \text{trophy},\ \text{obstacle}\}$, $\hat{X} = \{\emptyset,\ \text{left},\ \text{right}\}$, $\hat{Y} = \{\emptyset,\ \text{up},\ \text{down}\}$, $\hat{W} = \{\emptyset,\ R,\ L,\ UR,\ UL,\ F,\ B\}$ are the domains of $o$, $x$, $y$ and $w$ respectively. Notice that the relation FarFromObject$(a,$ trophy$)$ is also true for some adequate values for $m$ and $n$. Utilizing these relations, a collection of states was established to determine the agent's current state based on its interactions with the elements within the environment. It's worth noting that these relations were initially formulated to enable the agent to make decisions using straightforward spatial directions, thus simplifying the set of states.

### 5.3 Learning to Play River Raid

The agent underwent training in Atari's River Raid game utilizing the proposed algorithm across 500 episodes with with parameter settings of $\epsilon = 0.1$ and $\gamma = 0.6$, with periodic testing conducted every 50 episodes to assess the effectiveness of the policy generated up to that point. The outcome of this experiment is depicted in Figure 5. Notably, the experimental results exhibit an upward trend as the episodes progress, displaying a positive linear trajectory. The selection of the hyperparameters $\epsilon$ and $\gamma$ was guided by Machado et al.'s research [5], which compiles data from experiments conducted across various Atari games using the DQN algorithm, selected as a benchmark for comparison in this study.

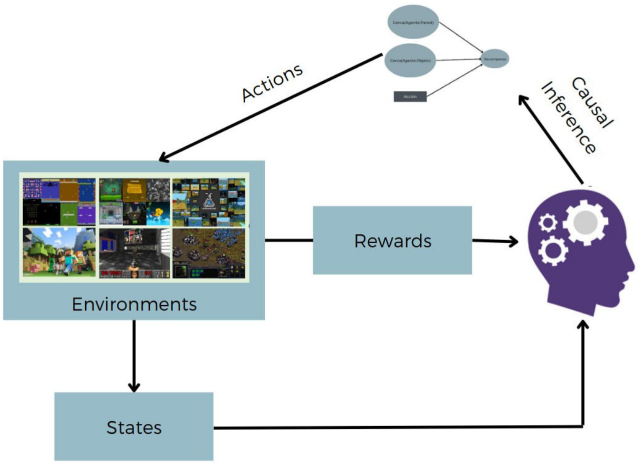


**Fig. 7.** Incorporation of causal models based on the relationships obtained from object detection for different similar tasks.

Table 2 presents the average scores obtained from 100 episodes using the proposed approach and a known algorithm used in Atari game environments [5]. Values in parentheses indicate the corresponding standard deviations. It can be seen that the RQ-Learning approach obtained comparable results with DQN in a smaller number of frames. While the outcomes between the suggested approach and DQN might exhibit similarities in certain scenarios, it's crucial to acknowledge that due to the stochastic nature of video games, these results can exhibit substantial variations. Additionally, it's important to consider the necessity of pre-training the object detection model when aiming for knowledge transfer, as this model must be adapted with information from the new environment before agent training can commence.

### 5.4 Transfer Learning

Building upon the previous results, the subsequent experiment involved applying the policy derived from 1000 episodes of gameplay in River Raid to another game. Chopper Command was selected for this task due to its shared objectives and similarities with River Raid. Initially, the agent attempted independent policy learning, followed by the application of a pre-existing policy to evaluate the agent's capacity to leverage this acquired knowledge effectively. This policy is structured as a Q-table and is deployed by the agent during the algorithmic step where it selects the most suitable relation during object detection.

Given that the relationships within the set of states remain consistent across both games, the transference of this environmental abstraction is feasible. From the results we can see in Figure 6, the agent starts with higher reward values when transfer is made. On the other hand, learning is more stable. These results lead us to conjecture that knowledge transfer using a policy learned in one game to another is possible if the environments have enough similar characteristics that can be represented with relations.

### 5.5 Discussion

The results obtained so far show us that the object detection system works adequately under the conditions of the reinforcement learning environment. This is due to the object detection system's ability to not only classify objects within a given image but also provide valuable geometric information about them. By leveraging this spatial information, it becomes possible to establish spatial relations between these objects and define a set of states for the agent.

Furthermore, in the case of the experiment in learning to play River Raid, it can be observed that the agent reaches a new maximum reward approximately every 100 - 150 episodes, so it can be conjectured that given more training time, the agent will find an optimal policy as the number of episodes increases. Also, the preliminary results in transfer learning suggest that knowledge transfer between games is possible when they share similar characteristics represented by relations.

## 6 Future Directions

Based on the relationships identified by the detection system, the proposed approach suggests using a causal model derived from these relationships to represent the immediate effect of an action on a given set of relationships as illustrated in Figure 7. The idea is that the agent learns a causal model at the same time it is learning a policy, based on this abstract relational representation. Once it has a partial model [8], it can use it in selecting the appropiate actions (for instance, those that maximize the reward), and thus accelerate even more the learning process. This is left as future work.

## 7 Conclusions

The use of relational representation and causal models can improve an agent's ability to understand complex environments, resulting in more efficient decision-making. This work proposes a way to incorporate an external sensing system into reinforcement learning environments, particularly in Atari games, and provides some initial findings on training the agent from relations.

It is expected that these results will lead to a way of transferring the knowledge acquired by the agent to similar tasks, thus defining an approach that combines elements of computational vision and relational representation with reinforcement learning. The results obtained from the proposed approach showed that it achieved results comparable to existing works (e.g., DQN) in the selected games. In particular, an improvement in learning speed and knowledge transfer between similar games was observed.

The proposed approach suggests using a causal model derived from the relationships identified by the detection system to represent the immediate effect of an action on a given set of relationships and the immediate reward. Through this study, we can speculate that this approach might enhance the agent's environmental understanding, resulting in more effective decision-making and knowledge transfer to tasks sharing a similar causal structure, facilitated by abstract environmental relationships.

## References

1. Ammanabrolu, P., Riedl, M.: Transfer in deep reinforcement learning using knowledge graphs. In: Proceedings of the 13th Workshop on Graph-Based Methods for Natural Language Processing, pp. 1–10 (2019) doi: 10.18653/v1/d19-5301
2. Bellemare, M. G., Naddaf, Y., Veness, J., Bowling, M.: The arcade learning environment: An evaluation platform for general agents. Journal of Artificial Intelligence Research, vol. 47, pp. 253–279 (2013) doi: 10.1613/jair.3912
3. Garnelo, M., Arulkumaran, K., Shanahan, M.: Towards deep symbolic reinforcement learning. pp. 1–13 (2016) doi: 10.48550/ARXIV.1609.05518
4. Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., Wierstra, D.: Continuous control with deep reinforcement learning. In: Proceedings of the International Conference on Learning Representations, pp. 1–14 (2016) doi: 10.48550/ARXIV.1509.02971
5. Machado, M. C., Bellemare, M. G., Talvitie, E., Veness, J., Hausknecht, M., Bowling, M.: Revisiting the arcade learning environment: evaluation protocols and open problems for general agents. Journal of Artificial Intelligence Research, vol. 61, no. 1, pp. 523–562 (2017) doi: 10.48550/ARXIV.1709.06009
6. Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., Hassabis, D.: Human-level control through deep reinforcement learning. Nature, vol. 518, no. 7540, pp. 529–533 (2015) doi: 10.1038/nature14236
7. Morales, E. F.: Scaling up reinforcement learning with a relational representation. In: Proceedings of the Workshop on Adaptability in Multi-Agent Systems, pp. 15-26 (2003)
8. Méndez-Molina, A., Feliciano-Avelino, I., Morales, E. F., Sucar, L. E.: Causal based Q-learning. Research in Computing Science, vol. 149, no. 3, pp. 95–104 (2020)
9. Raedt, L. D., Kersting, K., Natarajan, S., Poole, D.: Statistical relational artificial intelligence: Logic, probability, and computation. Synthesis Lectures on Artificial Intelligence and Machine Learning, 1st Edition (2016) doi: 10.1007/978-3-031-01574-8
10. Sutton, R. S., Barto, A. G.: Reinforcement learning: An introduction. The MIT Press (2020)
11. Tessler, C., Givony, S., Zahavy, T., Mankowitz, D., Mannor, S.: A deep hierarchical approach to lifelong learning in minecraft. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 31, no. 1, pp. 1553–1561 (2017) doi: 10.1609/aaai.v31i1.10744

ISSN 1870-4069

# Web Service for Inventory Control with Machine Learning and Interactive Maps with Leaflet on the Google Cloud Platform

Jaime A. García-Pulido[1], José M. Salomón-Montaño[1],
Uziel H. López-Meneses[2], Pablo Yamamoto-Magaña[1],
Raúl Morales-Salcedo[2], Yoel Ledo-Mezquita[1], Elioth Macias-Frotto[1]

[1] Tecnológico de Monterrey,
School of Engineering and Sciences,
Mexico

[2] NDS Cognitive Labs.,
Mexico

{a01652094, a01252989, a01733922, a01022382, yledo,
a00837025}@tec.mx, raulms@ndscognitivelabs.com

**Abstract.** This article will show and detail the process that was followed for the development of a web page capable of tracking delivery vehicle routes using interactive maps using Leaflet technology, and an inventory control optimized with Machine Learning for the calculation of replenishment of inventory in distribution centers. Copyright © 2022 TEC de Monterrey y NDS Cognitive Labs.

**Keywords.** Machine Learning, Leaflet, React, Node.js, Javascript, HTML, CSS, Highcharts.

## 1 Introduction

Due to the growth of establishments and companies responsible for delivering products to these places for restocking, and due to the large population growth, there has arisen the need to create a system that companies can use to keep a record of their shipments, both past and present, to ensure that the products sent are arriving on time and in good condition to their respective destinations.

In this article, we aim to describe the procedure followed to create two screens of a web application that allow us to keep track of orders and their frequency with an interactive map; and a screen capable of displaying inventory information from each distribution center optimized with Machine Learning to calculate when a new order should be created to supply the distribution centers responsible for these products.

We also seek to develop these two screens in the Visual Studio text editor with the Node.js and React tools along with the JavaScript programming language. All these concepts will be explained in the next section.

pp. 45–53; rec. 2023-06-04; acc. 2023-09-01

## 2 Definitions and Acronyms

Before starting with the process and steps that were followed to achieve our results, it is essential to mention the different concepts, definitions, and technologies that will be used throughout the entire article. This is to make reading this article easier to understand by managing these concepts.

### 2.1 Frontend

Refers to the visual part of the website. It is what displays when showing all the visible elements the website has. Users can not only view these elements but also interact with them, whether by clicking on them, moving with the tab key, or using the mouse wheel to scroll through the page.

### 2.2 Backend

The backend refers to everything that happens behind the page. All the processing and/or logic that may take place. This can be seen, for example, when entering an account and its respective password, the backend takes care of registering that information and verifying if the provided data is correct to then redirect the user to the main page or their personal account.

### 2.3 Database

All the information that can be obtained/retrieved from the website is stored in a database, from the username, passwords, or the settings of a specific user. Another example would be a bank account and how much money are in a particular account and to whom it belongs.

### 2.4 Visual Studio

This is the text editor that helps us develop the frontend and backend of our website. It was the text editor chosen for this project.

### 2.5 API

An environment that allows two software elements to communicate with each other for the application to function correctly. An example of this would be the weather application found on a phone. It communicates with a database that contains all the meteorological data, and the API allows us to have those updates on our phone. In this way, this information can be reinterpreted so that the phone can understand and update it [1].

### 2.6 HTML and CSS

The language is responsible for the distribution/skeleton of the various elements that make up the website. For instance, this language helps us create buttons, menus, whether navigation or side menus, etc. On the other hand, the CSS language helps us style the website, from the background color, the type and size of the font, border color, structure of components, etc. [2].

### 2.7 Javascript

This programming language helps us give logic to the website; it helps the backend function. It can also act as a bridge between HTML and CSS when using an API that can generate these 2 languages from Javascript to facilitate our application's development.

### 2.8 Node.js and React Node

The programming environment that allows us to run all our Javascript programs in Visual Studio. Within these programs, we will be using React, an API that helps us generate HTML and CSS codes directly from Javascript and allows us to create components for our website that can be reused in different parts of it [3].

### 2.9 Leaflet

An open-source visual tool that helps us embed maps on our website. It works similarly to Google Maps, but we chose Leaflet because it is free, unlike the tool Google offers. In the table below, we can observe the most significant differences between the two tools, considering the features deemed most important for this project's development.

Even though Google Maps has more features and functions than Leaflet, as seen in Table 1, the decisive factor was the price, as this way, the project's budget is safeguarded.

### 2.10 Highcharts

For the parts of the page that need a graph to display our inventory system data, we used the Highcharts library. This library allows us to modify our graph's title, the data we want to show, the axis names, etc. This library also lets us export the graphs to different file formats, such as image formats, .CSV file extensions, or even print the graph if desired.

### 2.11 Machine Learning

This term refers to one of the branches of artificial intelligence. This concept allows us to program computers to 'learn' to classify data and scenarios based on past experiences or information collected over time [4].

**Table 1.** Comparison between Google Maps and Leaflet.

| **Differences between APIs** | | |
|---|---|---|
| | Google Maps | Leaflet |
| Price | MXN $416 | Free |
| Markers | Yes | Yes. |
| Route | Yes | Yes, with an additional library. |
| Heat maps | Yes | Yes, with an additional component. |
| Documentation | Extensive | Moderate |

### 2.12 Google Cloud Platform

Finally, we have Google's cloud platform. This platform makes it easier for us to place our website on a secure domain provided by Google. Also, if we need any of its various APIs, Google's platform allows us easy access, but for now, we will only use it to host our website.

## 3 Development

At the beginning of the project, the first thing decided was the interactive map where the user can filter the cities or establishments they want to see on the map at any time. Due to the vast number of places in our data, from Boston to Rio de Janeiro, it is quite challenging to discern one establishment from another when having a global view on the map.

To solve this, we decided to use a heat map on the entire map to visualize the frequency with which each establishment appears in our data. In the heat map, a warm color, for example, red, indicates a more frequent presence of locations on the map, while a cooler color, like blue, indicates that there isn't as much frequency of that particular establishment, or it only appears once in our data.

All of the above was created using Leaflet as a tool for the interactive map, while for the development environment, Visual Studio was used as a text editor, and Node.js and React as APIs for the entire page and its components.

For this, 4 drop-down menus were created where the type of delivery that each establishment had, the type of vehicle used for each delivery, the product of each delivery for each establishment, and the establishment itself, filtering by their respective names, can be filtered. In this way, only what the user wants can be displayed on the map. This works best with the markers developed for this part of the project, so we have two ways to view the map. One would be with the heat map to visualize the frequency, and the other would be with individual markers to show every establishment.

Having accomplished the above, another screen was developed where inventory calculation is performed. This part of the website is responsible for keeping a record of the total inventory that the distribution centers have.

For this page, a dropdown menu was developed that allows the selection of a product from a particular inventory, this being the inventory of each distribution center. Hence, the person in charge of the inventory can only view the inventory assigned to them and not the inventory of another distribution center.

Therefore, only the current inventory being viewed appears on the page and cannot be modified. Once the product is selected, the user can view one of four charts available on the page. The default chart is the EOQ chart, which shows how much of a particular product remains in inventory. The other three charts display the quantity delivered with each restocking order, the price of each order over time, and the frequency of the orders. These charts can be seen in Figs. 1 to 4.

To change the desired chart for analysis, there are four buttons on the left side of the screen that display which chart is shown when clicking on one of these buttons. Additionally, if the user wishes to download the chart for further analysis, they can download the chart in image format or data tables to use them in another application or analysis.

There is also a button at the bottom of the screen for the user to place an order to restock the distribution center when the inventory chart indicates that an order is needed. For this part, the backend is equipped with Machine Learning to predict when is the best time to order more inventory.

In this case, the variables to be used for this prediction would be the time when things should be ordered, in other words, when the inventory is lower than a certain limit, and how long it takes the supplier to deliver the necessary products so that the inventory exceeds the previously mentioned limit.

## 4 Implementation

As previously mentioned, the text editor used for the development of the website was Visual Studio, and Node.js was used to utilize the React API [5]. For the map part, the dropdown menus were first developed, where the user can select the filters, they want to apply to the map. To get the different order options for the filters, data was obtained through a JSON file containing the geolocation data of each store, the type of order and product received, as well as the vehicle used for the delivery. Once this was achieved, it was verified that the JSON could be accessed correctly to proceed with the implementation of the interactive map.

### 4.1 Interactive Map with Leaflet

As previously mentioned, Leaflet was used as a tool for the map. To insert the map on the website, the component was first developed separately to verify the functionality of the tool and understand what features we could implement. In this case, markers and heat maps offered by the tool were decided to be used.

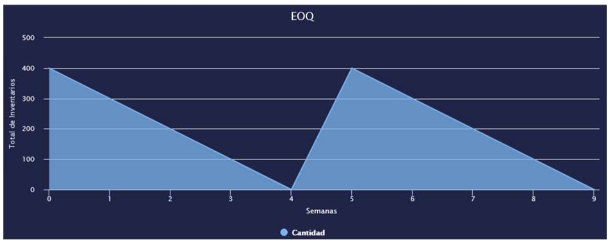


**Fig. 1.** EOQ Chart with Highcharts.

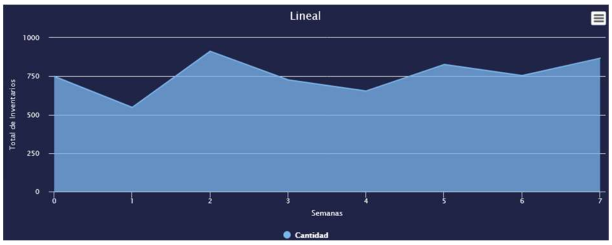


**Fig. 2.** Time series chart of inventories with Highcharts.

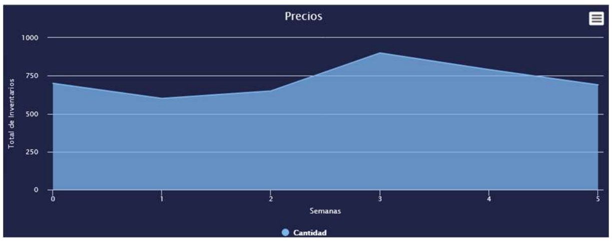


**Fig. 3.** Time series chart of prices with Highcharts.

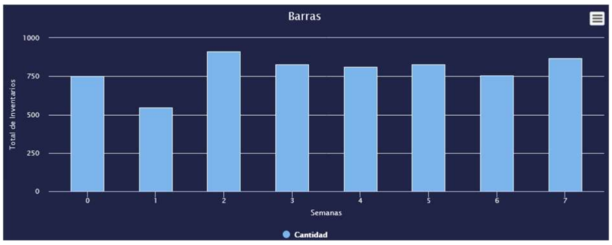


**Fig. 4.** Bar chart of inventories with Highcharts.

When implementing the map, the initial coordinates at which the map will position, when the map loads must first be provided to the tool, along with a number from 0 to 20 indicating the zoom level the map will have over the initial coordinates.

Then, for the implementation of the markers, it was only necessary to provide the longitude and latitude coordinates of each establishment, and if desired, a text could be added indicating the name of the establishment and the vehicle used in that delivery.

On the other hand, for the heat maps, it was necessary to provide the coordinates of the establishments, in the same way as the markers, and add them to the map using

Leaflet's heat Layer function and the leaflet. Heat component, which are responsible for loading these geographical points on the map to create heat maps. In this way, both functionalities in Leaflet can be implemented.

To apply the necessary filters to the map, it was only necessary to filter by name or type of delivery within the JSON and then display the location points that match the applied filters. If there's a match, it's displayed on the map; otherwise, the map remains empty.

### 4.2 Inventory Control with Machine Learning

Finally, for inventory control, the dropdown menu was first developed to select the product whose inventory is to be viewed at the moment. Then, the chart was implemented where the different inventory data can be viewed, for example, with an EOQ chart. Next, the buttons were created with which the type of chart to be viewed in the previously developed chart section can be selected. Lastly, a button was added that, as mentioned in section 3 of this document, will allow an order to be placed to restock the distribution center with the desired product.

For the chart part, the Highcharts library was used [6], which allows us to easily modify many of its functions. In this case, area charts were used for the EOQ chart, a time series chart for costs, a bar and time series chart indicating the quantity delivered with each order, and another time series chart showing how often orders have been made.

Thanks to the Highcharts library, it is easy to modify specific parts of each chart according to the needs of what is to be shown. In this case, mainly, the titles of the charts were modified, as well as the names of the x-axis and the y-axis, the names of our data, etc. This library also allows us to export any chart we want, either to an image, export it to a PDF file, or it can also be exported as a .CSV file, which allows us to reuse these data elsewhere, for example, for the analysis of orders, prices, among other things.

Finally, to modify any of these chart properties, an object must be created that has all the specifications and features for our charts.Once that is done, we simply have to specify which feature object we are going to use for each chart when creating the Highcharts component, in this way we can reuse code and decide when to show each of the charts.

When integrating this screen with the rest of the website, a request must be made to a file that contains the code responsible for the Machine Learning analysis to then show the request results on this screen so that the user can decide whether to place the order based on the results of the charts.

## 5 Results

Once the above development was achieved, it was possible to create two responsive screens depending on what the user desires. It was also possible to create each of the components described in this document with the tools mentioned in the previous sections. The following images show some sections of the screens that were created to create the inventory system website.

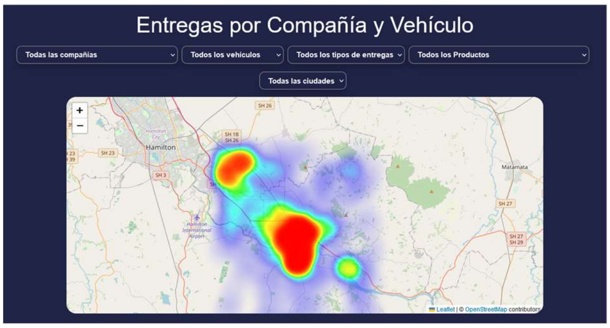


**Fig. 5.** Final screen of the interactive map with Leaflet.

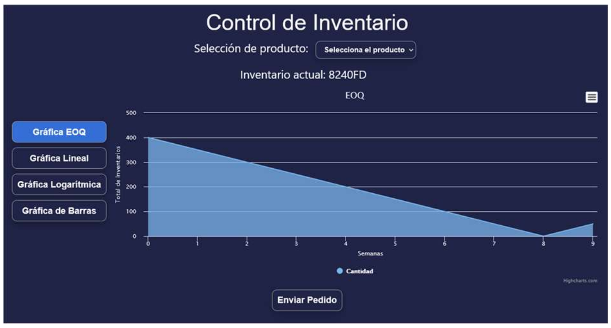


**Fig. 6.** Final screen of the inventory system.

As shown in Figs. 1 and 2, you can see how the screens of both applications run, taking into account the CSS styles assigned to each of the screens. In Fig. 1, we can observe the interactive map with Leaflet using an example of a heat map showing the frequency with which some of the establishments are reached for deliveries. In Fig. 2, you can see the buttons and the dropdown menu implemented to change the charts you want to view.

## 6 Conclusions

Thanks to the procedure that was followed, it was possible to develop the two screens and their functionalities correctly. Initially, it was a challenge and confusing to learn a new API and a completely new development environment. Fortunately, there is a lot of documentation regarding these two technologies, and in the end, if there is any doubt about how to make certain implementations or functions in an application, there is a solid documentation base where a possible solution can be sought.

Regarding the development of the screens, thanks to the technologies learned to generate these applications, it was possible to implement the necessary functionalities for the screens to function correctly.

For the interactive map part, it was a challenge to embed the map on the page and have everything work correctly on the user's side, this being the logical part of that page. Plus, it was a challenge to learn how the Leaflet tool worked to take full advantage of its potential. Finally, for the inventory control part, the High charts tool was learned to implement each of the necessary charts so that the user can interpret the data displayed on the screen, and so that the user can download or export those charts and can handle this data more easily on their own. Fortunately, thanks to all the previous learning acquired when developing the interactive map, it was beneficial to develop this part of the web application.

The choice of such tools (leaflet and High charts) allows for efficiency and effectiveness, enabling companies to remain competitive. This study lays the groundwork for future research and developments on the integration of these technologies into logistical and inventory systems.

## References

1. How to create a public API with AWS. https://aws.amazon.com/es/what-is/api/
2. NA: Hypertext Tag Language | MDN. https://developer.mozilla.org/es/docs/Web/HTML (2022)
3. Leaflet: An open-source JavaScript library for interactive maps. https://leafletjs.com/ (2022)
4. Alameda, T.: Machine Learning: What is it and how does it work? BBVA NEWS, https://www.bbva.com/es/machine-learning-que-es-y-como-funciona/ (2022)
5. Jones, M.: Creating a Node.js and React application - Visual Studio (Windows). https://learn.microsoft.com/es-es/visualstudio/javascript/tutorial-nodejs-with react-and-jsx?view=vs-2022 (2022)
6. Jhenrryalvaro: HighCharts: Library for creating charts. Accessed: Available: https://enboliviacom.wordpress.com/2013/03/01/highcharts-libreria-para-creacion-de-graficos/ (2022)

ISSN 1870-4069

# Enhancing Biomedical NLP in Spanish: Large Language Model's Domain Adaptation for Named Entity Recognition in Clinical Notes

Rodrigo del Moral, Orlando Ramos-Flores,
Helena Gómez-Adorno

Universidad Nacional Autónoma de México,
Instituto de Investigaciones en Matemáticas
Aplicadas y en Sistemas,
Mexico

rodrigodelmoral@comunidad.unam.mx,
orlando.ramos@aries.iimas.unam.mx,
helena.gomez@iimas.unam.mx

**Abstract.** Named Entity Recognition (NER) plays a crucial role in extracting valuable information from clinical texts, enabling the organization of relevant data within databases and knowledge bases. However, automatic recognition of named entities in clinical notes poses significant challenges due to the diversity of writing styles and vocabularies. This paper investigates domain adaptation techniques' effectiveness of large language models in enhancing NER in Spanish clinical texts. We propose using pre-trained language models and a corpus of Mexican clinical notes, applying anonymization techniques to protect sensitive information. The study evaluates the performance of adapted models on publicly available NER datasets, Cantemist and PharmaCoNER. Results indicate a slight improvement in NER performance, showcasing the potential of domain adaptation to tailor language models for specific clinical domains. Future research directions are discussed to refine domain adaptation strategies and promote the responsible use of AI in healthcare applications.

**Keywords:** Large language models, domain adaptation, clinical texts, named entity recognition.

## 1 Introduction

The concept of a "Named Entity" refers to the mention of real-world objects within a text, which can be made using common nouns or proper names. In the context of clinical texts, named entities can encompass references to individuals, locations, diseases, symptoms, medications, treatments, among others.

The ability to identify and extract named entities from clinical texts has become increasingly relevant, as it enables the organization of relevant information within databases, such as electronic health records (EHR). When extracted accurately, this information can be linked to knowledge bases, such as taxonomic, diseases, and pharmaceutics databases. Various applications have been proposed for clinical information databases, including the development of tools for automated

pp. 55–65; rec. 2023-06-04; acc. 2023-09-01

pre-diagnostic assistance [8] and systems for detecting and monitoring medication side effects [17]. More specific applications have also been suggested, such as models for monitoring and predicting suicidal crises in vulnerable patients [13]. These proposals seek to leverage machine learning algorithms to generate knowledge but heavily rely on carefully extracted data for effective training. While these applications hold promise, building models that can automatically recognize, classify, and extract named entities is no trivial task. The challenges largely stem from the inconsistencies in the clinical notes under analysis. These notes, authored by different physicians from various specialties and backgrounds, exhibit an overwhelming diversity of structures, vocabularies, and styles.

Consequently, rule-based model creation becomes prohibitively excessively time-consuming and resource-intensive, requiring many rules that work in concert and require validation by medical experts [4]. Several language models based on machine learning algorithms have made advancements in this field. However, these models, such as Med-BERT [16] and BioBERT [9], have been primarily trained on English medical texts.Additionally, tools have emerged, such as IBM Watson Health, Google Cloud Healthcare API, and MS Azure Text Analytics for Health, that provide complete pipelines for recognizing certain general types of clinical entities in English.

However, progress in developing specific models for clinical texts in the Spanish language has been limited. A key reason behind this disparity is the scarcity of clinical texts in Spanish for training these models; to build systems akin to those in English, an equally substantial dataset would be required. An alternative approach to creating a language model for a specific domain as clinical texts in Spanish is domain adaptation [7]. This technique involves taking a base model pre-trained on a general domain corpus (i.e Spanish texts) and then retraining it with text from a more narrowly defined target domain (i.e., clinical texts). The volume of training data needed in domain adaptation is not as large, since the base model has already learned certain language features during its pre-training [18]. Hence, by starting with a properly pre-trained language model that has been adapted to the desired domain, training a model capable of recognizing entities specific to the clinical domain should yield superior results [5].

In this paper, we investigate the effectiveness of domain adaptation techniques for named entity recognition in clinical texts using pre-trained language models in the Spanish language. We explore the challenges associated with the scarcity of clinical data in Spanish and propose strategies to enhance the performance of named entity recognition models for Spanish clinical texts. The rest of the paper is structured as follows. Section 2 provides a review of the current literature on language models and their performance in biomedical and clinical NLP tasks.

Section 3 details the creation of a clinical notes corpus along with anonymization techniques. Section 4 discusses the selection of RoBERTa-based models and the domain adaptation process. Section 5 explains the setup for evaluation using the Cantemist and PharmaCoNER datasets. Section 6 presents the performance of language models in NER tasks, comparing different base models with domain-adapted models. Section 7 highlights the potential of domain adaptation and outlines future research directions. The paper ultimately offers insights into adapting language models for specific domains and improving NER in clinical texts.

## 2 Related Work

In the study conducted by [11], a comprehensive set of experiments was carried out using diverse datasets, spanning both scientific and clinical domains, to address common modeling tasks. These included Named Entity Recognition (NER) tasks as well as text anonymization (also referred to as de-identification). In addition, to tasks that involved relation extraction, multi-class and multi-label classification, and Natural Language Inference (NLI)-style tasks.

The authors compared five publicly-available language models to gain a representative understanding of the state-of-the-art in biomedical and clinical NLP. Additionally, they pre-trained new models on their curated corpora and examined key design factors that affect downstream performance in BioNLP tasks. Specifically, they focused on three criteria: i) the impact of model size on downstream performance, ii) the influence of the pre-training corpus on downstream performance, and iii) the significance of tokenizing with a domain-specific vocabulary on downstream performance.

Their experimental approach largely followed the pre-training methodology of [12]. The findings revealed that RoBERTa-large consistently outperformed RoBERTa-base, despite both models having access to the same training corpora. Furthermore, among the publicly available models they experimented with, BioBERT demonstrated the best performance. The newly introduced models in their study exhibited strong performance, achieving superior results on 17 out of the 18 tasks compared to the existing models, often by a significant margin.

Moreover, a team of researchers from the Barcelona Supercomputing Center (BSC) addressed the language gap in Spanish by pre-training two Transformer-based language models from scratch [2]. They compiled biomedical and clinical corpora of various sizes and sources. This included an Electronic Health Record (EHR) corpus containing 95M tokens from over 514k clinical cases, as well as a biomedical corpus with 1.1B tokens across 2.5M documents. The RoBERTa base architecture, featuring 12 self-attention layers, was employed for pre-training, focusing solely on Masked Language Modeling (MLM) using Subword Masking (SWM) as the training objective.

Tokenization was done with the Byte-Pair Encoding (BPE) algorithm, resulting in a vocabulary of 50,262 tokens. The authors produced two RoBERTa models: bsc-bio-es, trained with biomedical resources only, and bsc-bio-ehr-es, which utilized both the biomedical and the EHR corpus. They evaluated the models by fine-tuning for Named Entity Recognition (NER) on three distinct datasets: PharmaCoNER [3], CANTEMIST [14], and ICTUSnet[1].

When comparing their models with various benchmarks —including general domain and domain-specific models— they achieved significant improvements over general-domain models and matched or outperformed domain-specific models across all tasks.

[1] ictusnet-sudoe.eu/es/

## 3 Pre-Training Corpora

In order to perform domain adaptation on pre-trained models, we compiled a corpus from a set of clinical notes penned by doctors based in Mexico City. This corpus was initially comprised of 86,392 notes taken from Electronic Health Records (EHRs), wich contained records mentioning the Interrogation, Physical Examination, Symptoms, Vital Signs, Auxiliary Studies, Diagnosis, Management Plan, and Treatment Plans of patients.

However, a significant portion of these clinical notes contained irrelevant information; for instance, many fields within the notes were filled with words such as “empty”, “no”, and similar. Moreover, these notes held sensitive information, which must be anonymized to be used in the training of a model without jeopardizing the privacy of those involved [6]. It should be noted that around 98% of these notes originate from emergency services within Mexican hospitals, a factor that merits consideration when analyzing the performance of models trained with this data.

Consequently, we designed a pre-processing algorithm comprising several steps. Initially, we eliminated all notes that solely contained vital signs measurements. Additionally, we removed discharge and follow-up notes that contained repeated or redundant text to circumvent potential memorization issues [10].

To anonymize any names found within the notes, we developed an algorithm that consistently substitutes these. We retrieved names from both the ‘patient’s name’ and ‘doctor’s name’ fields in the clinical notes database, and subsequently generated sets from these. We supplemented these sets using lists of common Spanish names provided by public statistical services institutes in both Mexico and Spain. These produced sets distinguished between predominantly male and female names, and also included a category for gender-neutral names. The resulting anonymization sets included 468 male names, 563 female names, 97 gender-neutral names, and 1130 surnames. The discrepancy in the number of names and surnames can be attributed to the fact that compound names were split and then handled individually.

Following the creation of the names sets, we begin the anonymization process. The algorithm processes each note, normalizing it to lowercase and stripping away accents and special characters. It then seeks exact matches with each name and surname present in the names sets. If a match is found, it is substituted, preserving the accents and special characters and adapting it to match the original case. For each name that is matched, the substitution is recorded and applied consistently to any subsequent matches in the same note. Additionally, to account for possible spelling errors, a search using Levenshtein distance is performed for the correct patient’s and doctor’s names in the original database. Moreover, the algorithm to generate random names is designed not to repeat names within the same note.

As part of our analysis of the clinical notes, we found instances where doctors had recorded sensitive personal information, such as the patient’s address, phone number, and other potentially sensitive data. Consequently, any lines of text containing phone numbers, email addresses, and physical addresses were eliminated. We also masked any alphanumeric codes that could potentially be identifiers, such as patient numbers, professional IDs, and social security numbers.

At the end of this process, we had amassed a corpus of 78,616 unique and anonymized clinical notes. We identified and anonymized 47,237 names and 63,982 surnames, with at least one present in 80% of the notes. We discarded 244 phone numbers and alphanumeric codes, as well as 1,697 addresses. The final token count for the retraining process is 39,680,624.

## 4 Domain Adaptation of Models

To generate an adapted language model tailored to the domain of Mexican clinical notes, we chose models based on the RoBERTa architecture as a starting point. This architecture is preferred for domain-specific model training because it facilitates the learning of new vocabularies through the Byte Pair Encoding (BPE) tokenization algorithm [15]. Specifically, we started by selecting the most prominent language model for the biomedical domain in Spanish as our base-model: the RoBERTa-based bsc-bio-ehr-es[2]. This model achieves the best results for Named Entity Recognition (NER) tasks in the Spanish biomedical domain using a simple fine-tuning architecture [2].

Even though the BSC model is trained entirely on biomedical texts, only 16.17% of the training corpus belongs to the clinical domain. Out of this percentage, 48% of the tokens belong to clinical notes; and the remaining tokens in the clinical domain are comprised of 'clinical cases', which fall within the same domain of knowledge but differ slightly in writing style. These differences in writing style, vocabulary, and carried-out medical procedures suggest that enriching the training process with the Mexican clinical notes corpus could enhance model performance for tasks specifically dealing with such texts. This approach is advantageous not only for addressing the unique attributes present in Mexican clinical notes but also for capturing the specific nuances inherent to Spanish-language clinical text.

To get the Mexican clinical notes ready for training, we set up a careful preparation process. First, we split the notes into batches of 512 tokens each. Then, we performed the tokenization using the BPE tokenizer from the BSC model trained with biomedical text. Each batch was generated following the full-sentence scheme described in the original RoBERTa article [12], in which the batches are filled with tokens from complete sentences regardless of crossing the document boundary. In total, 39,680,624 tokens from 78,616 notes were obtained for model training. The original training parameters from the BSC model were maintained, except for the batch size, which was reduced to 32 samples. No warm-up steps were performed; we started training with a rate of 5E-5 which was linearly decreased to 0.

The training was conducted over five epochs, saving a checkpoint at the end of each epoch to track the progress of the model. The retraining process took approximately 4 hours on two RTX-A5000 cards. Additionally, we carried out the same retraining process using the Mexican clinical notes corpus in two additional separate instances: first by leveraging a model[3] that had been previously fine-tuned from the original BSC model using the Chilean Waiting List (CWL) corpus [1]; and second, starting the

[2] huggingface.co/PlanTL-GOB-ES/roberta-base-biomedical-clinical-es

[3] huggingface.co/plncmm/roberta-clinical-wl-es

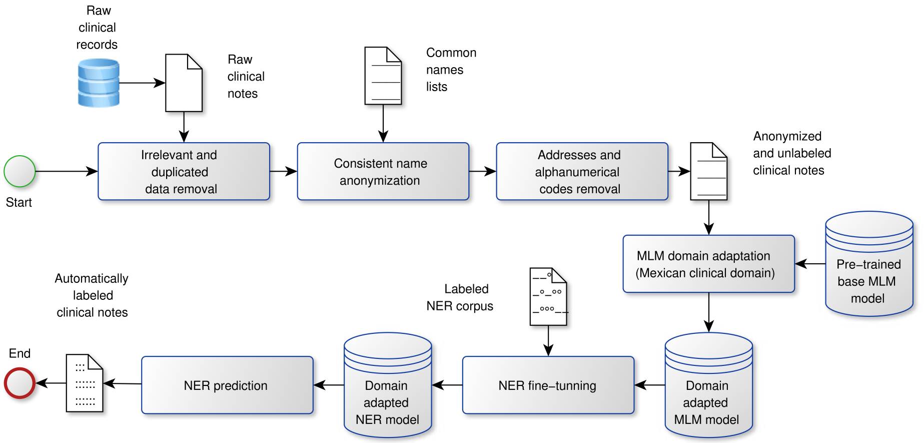

Raw clinical records
Raw clinical notes
Common names lists
Start
Irrelevant and duplicated data removal
Consistent name anonymization
Addresses and alphanumerical codes removal
Anonymized and unlabeled clinical notes
MLM domain adaptation (Mexican clinical domain)
Pre–trained base MLM model
Automatically labeled clinical notes
Labeled NER corpus
End
NER prediction
Domain adapted NER model
NER fine–tunning
Domain adapted MLM model

**Fig. 1.** General algorithm of the paper's code implementation.

multilingual general-purpose model, xlm-roberta-base[4]. Additonally to this description, in Figure 1, we show the complete pipeline of both the retraining of the MLM model using Mexican clinical notes, and the NER training step for evaluating the model.

## 5 NER Task and Datasets

We evaluated the performance of the domain-adapted language models on two Clinical Named Entity Recognition (NER) task. The evaluation was carried out using publicly available tagged datasets, the PharmaCoNER set[5] (Drugs and chemical entities) and the Cantemist set[6] (Tumors). This evaluation datasets were chosen due to the similarity of the texts they contain to the texts we used to retrain the model on the MLM task.

However, it is important to note there are still notable differences between the domains of the retraining texts and the NER evaluation datasets. While the retraining was grounded in a corpus of Mexican clinical notes primarily derived from real emergency episodes in hospitals — generally characterized by a certain urgency and spontaneity in documentation — the NER datasets exhibit more structured, meticulously crafted notes stemming from a diverse array of medical specialties.

This difference underscores a potential variation in complexity and style between the training and evaluation materials, and should be considered in the interpretation of the evaluation results. The PharmaCoNER set consists of 1,000 clinical cases written in Spanish and extracted from the Spanish Clinical Case Corpus (SCCC)[7], from a variety of clinical areas such as oncology, urology, cardiology, pneumology, among others. These clinical cases are annotated with 4 categories of entities: Normalizable Chemicals, Non-normalizable Chemicals, Proteins, and Others.

[4] huggingface.co/xlm-roberta-base

[5] huggingface.co/datasets/PlanTL-GOB-ES/pharmaconer

[6] huggingface.co/datasets/PlanTL-GOB-ES/cantemist-ner

[7] github.com/PlanTL-SANIDAD/SPACCC

**Table 1.** Results of the NER model trained with the Cantemist dataset. Precision (P), Recall (R), and F1 (F) scores on each dataset are reported. The best scores are in bold.

| **Cantemist** | | | |
|---|---|---|---|
| **Model** | **P** | **R** | **F** |
| bsc-bio-ehr-es | 0.8142 | 0.8581 | 0.8356 |
| bsc-bio-ehr-es + cwl | 0.8280 | **0.8570** | 0.8423 |
| bsc-bio-ehr-es + ours | **0.8295** | 0.8565 | **0.8427** |
| bsc-bio-ehr-es + cwl + ours | 0.8194 | 0.8509 | 0.8349 |
| xlm-roberta-base | 0.7958 | 0.8467 | 0.8205 |
| xlm-roberta-base + ours | 0.7995 | 0.8373 | 0.8179 |

Normalizable Chemicals refer to chemicals that can be linked to the SNOMED-CT knowledge base. The corpus contains 7,624 tagged entities in total. The Cantemist set consists of 1,301 clinical cases extracted from SCCC but exclusively from the oncology area. These clinical cases have been manually annotated by specialist doctors, identifying all mentions of cancerous tumors in one category: tumor morphology.

The dataset is balanced considering the age and gender of the patients, as well as the types of tumors. The corpus contains 16,030 tagged entities in total. We used a simple architecture to generate the NER model from the pre-trained RoBERTa model. A linear classification layer and a Softmax function were added after the output of the original model's hidden states. This model was trained for 10 epochs with an initial learning rate of 5E-5 and a linear decay rate. The batch size for training was set to 16 samples.

## 6 Evaluation and Results

Performance of the trained language models was evaluated in the NER task with the Cantemist and PharmaCoNER datasets. In addition, results from the base BSC language model (bsc-bio-ehr-es), the base XLM-RoBERTa (xlm-roberta-base), and the model enriched with the CWL notes (bsc-bio-ehr-es + cwl) are included for comparison. To measure the effectiveness of these NER models, we employed three universally acknowledged scores for NER tasks: precision, recall, and F1 score.

Precision indicates the proportion of correctly identified entities among all the entities that the model labeled, although it disregards the false negatives. In contrast, recall accounts for the proportion of correctly identified entities out of all actual entities, overlooking false positives. The F1 score is the harmonic mean of precision and recall, serving as a single metric that reflects a good balance of both.

All of these metrics were obtained using the micro-averaging method, thus giving equal weight to each entity in the final scores. These results can be found in Table 1 and Table 2, where the highest scores for each metric are highlighted to facilitate the comparison between models. The results in Table 1 show a slight improvement in the model's performance in the Cantemist task, which deals with finding mentions of cancerous tumors.

**Table 2.** Results of the NER model trained with the PharmaCoNER dataset. Precision (P), Recall (R), and F1 (F) scores on each dataset are reported. The best scores are in bold.

| **PharmaCoNER** | | | |
|---|---|---|---|
| **Model** | **P** | **R** | **F** |
| bsc-bio-ehr-es | **0.8921** | 0.9081 | 0.9000 |
| bsc-bio-ehr-es + cwl | 0.8703 | 0.9119 | 0.8906 |
| bsc-bio-ehr-es + ours | 0.8826 | 0.9119 | 0.8970 |
| bsc-bio-ehr-es + cwl + ours | 0.8876 | **0.9146** | **0.9009** |
| xlm-roberta-base | 0.8541 | 0.8788 | 0.8663 |
| xlm-roberta-base + ours | 0.8651 | 0.8891 | 0.8769 |

The model that stands out in this task is the one that was retrained with our dataset from the base BSC model (bsc-bio-ehr-es + ours) which presents a slight improvement in the precision score and F1 score. On the other hand, the bsc-bio-ehr-es + cwl model improves the performance slightly the recall score and in general is the best model compared to the base bsc-bio-ehr-es model. However, we note that when this model is subjected to an additional retraining stage (bsc-bio-ehr-es + cwl + ours), the performance drops slightly.

For this task, the worst models were the xlm-roberta-base and xml-roberta-base + ours. On the other hand, in the results described in Table 2 of the PharmaCoNER task, the performance slightly decreases when the base model is retrained, independently of the corpora. However, with the language model that has went through two sequential adaptations (bsc-bio-ehr-es + cwl + ours), the NER performance gets a minor uplift. One hypothesis for the drop in performance is that in the PharmaCoNER clinical cases, chemicals such as proteins are mentioned, which are not mentioned frequently within the clinical notes of our dataset.

Interestingly, this decline in performance dissipates when training incorporates data from both additional sources. Another reason is the clinical notes used for training our language model primarily belong to emergency services (98.54%) and the rest of the services (1.46%) are underrepresented. Besides, these notes lack normalization regarding typographical error correction (incorrect typing, omissions, duplications, inversions, replacements, white space omissions, accent omissions, and punctuation errors). We drop normalization at this particular point due to two reasons.

The first reason is the complexities of the medical domain, where technical terms, acronyms, abbreviations, and specialized jargon are prevalent, and addressing these linguistic intricacies requires expert knowledge. The second and more relevant reason is that we intend to use the trained LM to address downstream tasks with the same type of data, i.e., clinical medical notes direct from the real world.

Regarding the retraining of the xlm-roberta-base model, we observe a modest performance boost in the PharmaCoNER task when adapted to our corpus domain; though the results in the Cantemist task do not exhibit the same consistency. These fluctuations could potentially be attributed to the changes in the vocabulary size introduced by the multilingual model.

The vocabulary size of the xlm-roberta-base tokenizer is nearly five times larger than that of the BSC tokenizer, and has been trained on text spanning over 100 languages. As such, there's a compelling case for additional experimentation with models of this magnitude to uncover more detailed insights.

## 7 Conclusions and Future Work

This research paper analyzes the efficacy of domain adaptation techniques to enhance named entity recognition (NER) in Mexican Spanish clinical texts using pre-trained language models. Even though the downstream tasks for evaluation were not specifically tailored using the same specific domain (i.e. Mexican Clinical Notes), we observed an improvement in the performance on the NER tasks. This highlights the value of customizing MLMs to align them to the unique characteristics and nuances of the target domain.

The approach leverages RoBERTa-based language models and a corpus of Mexican clinical notes, which are anonymized to protect sensitive information. The study finds that domain adaptation through retraining on clinical notes can lead to a slight improvement in NER performance. Specifically, the model retrained with the generated clinical notes corpus shows promising results in recognizing cancerous tumor mentions in the Cantemist dataset.

For the PharmaCoNER dataset, an increased recall hints at the model's enhanced ability to extract more entities, potentially leveraging richer context from the clinical domain, albeit at the expense of precision; a decrease in the latter might suggest that while the model has become adept at analyzing clinical texts, it may still lack sufficient training data representing specialized biochemical substances, which are less frequently mentioned in the emergency notes dominant in our corpus.

The results suggest the potential of domain adaptation for refining language models to specific clinical domains and demonstrate the significance of sufficient domain-specific data for optimal performance. Furthermore, we examined the retraining of a large language model, specifically the xlm-roberta-base model, but the results were slightly inconsistent. This may be due to the considerable change in size from the model's pre-training corpus.

Future research directions are highlighted to explore more sophisticated domain adaptation techniques and to further refine the model's capabilities for NER in clinical texts. For the validation phase, we want to use downstream tasks based on texts from the specific domain we adapted, i.e. Mexican clinical notes. We also want to generate more diverse sets of evaluation tasks, both in the types of entities in NER, and in the variety of downstream tasks.

Additionally, we aim to conduct experiments on performance variations when adapting larger, multilingual models like the xlm-roberta-base model. Finally, as we see more AI systems being developed for healthcare applications, we must continue creating better ways to protect sensitive data. And as such, we want to continue improving our anonymization algorithms to ensure that healthcare data is used responsibly in our AI research.

**Acknowledgments.** This work has been carried out with the support of DGAPA UNAM-PAPIIT project number TA101722, support of Secretaría de Educación, Ciencia, Tecnología e Innovación de la Ciudad de México (Resource Allocation Agreement SECTEI/201/2021) in collaboration with Secretaría de Salud de la Ciudad de México (SEDESA), DGAPA-UNAM postdoctoral scholarship, and the CONAHCYT scholarship program (CVU: 1148113). The authors also thank CONAHCYT for the computing resources provided through the Deep Learning Platform for Language Technologies of the INAOE Supercomputing Laboratory.

## References

1. Báez, P., Villena, F., Rojas, M., Durán, M., Dunstan, J.: The chilean waiting list corpus: A new resource for clinical named entity recognition in spanish. In: Proceedings of the 3rd Clinical Natural Language Processing Workshop, pp. 291–300 (2020) doi: 10.18653/v1/2020.clinicalnlp-1.32
2. Carrino, C. P., Llop, J., Pàmies, M., Gutiérrez-Fandiño, A., Armengol-Estapé, J., Silveira-Ocampo, J., Valencia, A., Gonzalez-Agirre, A., Villegas, M.: Pretrained biomedical language models for clinical NLP in spanish. In: Proceedings of the 21st Workshop on Biomedical Language Processing, pp. 193–199 (2022) doi: 10.18653/v1/2022.bionlp-1.19
3. Gonzalez-Agirre, A., Marimon, M., Intxaurrondo, A., Rabal, O., Villegas, M., Krallinger, M.: Pharmaconer: Pharmacological substances, compounds and proteins named entity recognition track. In: Proceedings of the 5th Workshop on BioNLP Open Shared Tasks, pp. 1–10 (2019) doi: 10.18653/v1/D19-5701
4. Gorinski, P. J., Wu, H., Grover, C., Tobin, R., Talbot, C., Whalley, H., Whiteley, W., Alex, B.: Named entity recognition for electronic health records: A comparison of rule-based and machine learning approaches. In: Second UK Healthcare Text Analytics Conference (2019) doi: 10.48550/arXiv.1903.03985
5. Grangier, D., Iter, D.: The trade-offs of domain adaptation for neural language models. In: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics, vol. 1, pp. 3802–3813 (2022) doi: 10.18653/v1/2022.acl-long.264
6. Ishihara, S.: Training data extraction from pre-trained language models: A survey. In: Proceedings of the 3rd Workshop on Trustworthy Natural Language Processing, pp. 260–275 (2023)
7. Kalyan, K. S., Rajasekharan, A., Sangeetha, S.: AMMU: A survey of transformer-based biomedical pretrained language models. Journal of Biomedical Informatics, vol. 126 (2022) doi: 10.1016/j.jbi.2021.103982
8. Latif, J., Xiao, C., Tu, S., Rehman, S. U., Imran, A., Bilal, A.: Implementation and use of disease diagnosis systems for electronic medical records based on machine learning: A complete review. IEEE Access, vol. 8, pp. 150489–150513 (2020) doi: 10.1109/ACCESS.2020.3016782
9. Lee, J., Yoon, W., Kim, S., Kim, D., Kim, S., So, C. H., Kang, J.: BioBERT: A pre-trained biomedical language representation model for biomedical text mining. Bioinformatics, vol. 36, no. 4, pp. 1234–1240 (2020) doi: 10.1093/bioinformatics/btz682
10. Lee, K., Ippolito, D., Nystrom, A., Zhang, C., Eck, D., Callison-Burch, C., Carlini, N.: Deduplicating training data makes language models better. In: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics, vol. 1, pp. 8424–8445 (2022) doi: 10.18653/v1/2022.acl-long.577

11. Lewis, P., Ott, M., Du, J., Stoyanov, V.: Pretrained language models for biomedical and clinical tasks: Understanding and extending the state-of-the-art. In: Proceedings of the 3rd Clinical Natural Language Processing Workshop, pp. 146–157 (2020) doi: 10.18653/v1/2020.clinicalnlp-1.17
12. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V.: RoBERTa: A robustly optimized BERT pretraining approach (2019) doi: 10.48550/arXiv.1907.11692
13. Metzger, M. H., Tvardik, N., Gicquel, Q., Bouvry, C., Poulet, E., Potinet-Pagliaroli, V.: Use of emergency department electronic medical records for automated epidemiological surveillance of suicide attempts: A French pilot study. International Journal of Methods in Psychiatric Research, vol. 26, no. 2 (2016) doi: 10.1002/mpr.1522
14. Miranda-Escalada, A., Farré-Maduell, E., Krallinger, M.: Named entity recognition, concept normalization and clinical coding: Overview of the cantemist track for cancer text mining in spanish, corpus, guidelines, methods and results. In: Proceedings of the Iberian Languages Evaluation Forum, pp. 303–323 (2020)
15. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I.: Language models are unsupervised multitask learners (2019)
16. Rasmy, L., Xiang, Y., Xie, Z., Tao, C., Zhi, D.: Med-BERT: Pretrained contextualized embeddings on large-scale structured electronic health records for disease prediction. npj Digital Medicine, vol. 4, no. 1, pp. 1–13 (2021) doi: 10.1038/s41746-021-00455-y
17. Shinozaki, A.: Electronic medical records and machine learning in approaches to drug development. Artificial Intelligence in Oncology Drug Discovery and Development (2020) doi: 10.5772/intechopen.92613
18. Wiese, G., Weissenborn, D., Neves, M.: Neural domain adaptation for biomedical question answering. In: Proceedings of the 21st Conference on Computational Natural Language Learning, pp. 281–289 (2017) doi: 10.18653/v1/K17-1029

ISSN 1870-4069

# Optimizing Affordably Tree Seeding to Enhance Mexico City's Air Quality

Fernando Moreno, Adriana Lara

Instituto Politécnico Nacional,
Escuela Superior de Física y Matemáticas,
Mexico

alaral@ipn.mx, fmorenog0901@alumno.ipn.mx

**Abstract.** High levels of air pollutants represent a problem in many cities worldwide. Reforesting urban areas is a straightforward but challenging solution since different factors, such as pollen release and optimal location, should be considered. In this work, we propose a model to improve air quality in Mexico City via urban reforestation. A specific genetic algorithm was designed to obtain an optimal tree distribution over Mexico City counties. We provide an affordable solution to the problem, assuming a limited number of available seeding trees. A prediction model is also proposed for the solutions using publicly available data on the city's air quality and meteorological aspects. This work shows these numerical tools' potential to implement solutions to increase life quality based on publicly available data.

**Keywords:** Evolutionary algorithms, air quality, optimization.

## 1 Introduction

Unacceptable air quality is a significant issue in many cities all over the world [11]. Identified causes of bad air quality are urban extension and the population growth in developing countries [9]. In response, cities have launched numerous projects, including monitoring, reforestation, clean energies, awareness, etc [16]. Some of these ideas have worked, and much pollution has been reduced [21]. Over the years some evidence emerged that vegetation is important in reducing pollution concentrations [9].

Air pollution in Mexico City is conformed by several elements, i.e., Carbon Dioxide ($CO_2$), Ozone ($O_3$), Nitrogen Dioxide ($NO_2$), particles smaller than 10 micron ($PM_{10}$) and particles smaller than 2.5 micron ($PM_{2.5}$). Various human-related processes cause polluting emissions into the air; for example, industry, electricity generation, commerce, domestic sources of energy, and motorized vehicles [19].

Air-suspended particles are a complex mix of liquid and solid materials [14]; depending on their origin, they can vary in size, shape, and composition. The size of pollutant particles can vary in diameter from 0.005 to 100 microns [25]; their sources are diverse, such as pollen, plant emissions, and plastic burning, among others [4, 20]. Epidemiology studies have shown a correlation in urban areas between air-suspended particles, such as $PM_{2.5}$ and $PM_{10}$, in daily deaths and hospital entrees for cardio and respiratory diseases [3, 12, 25].

pp. 67–81; rec. 2023-06-04; acc. 2023-09-01

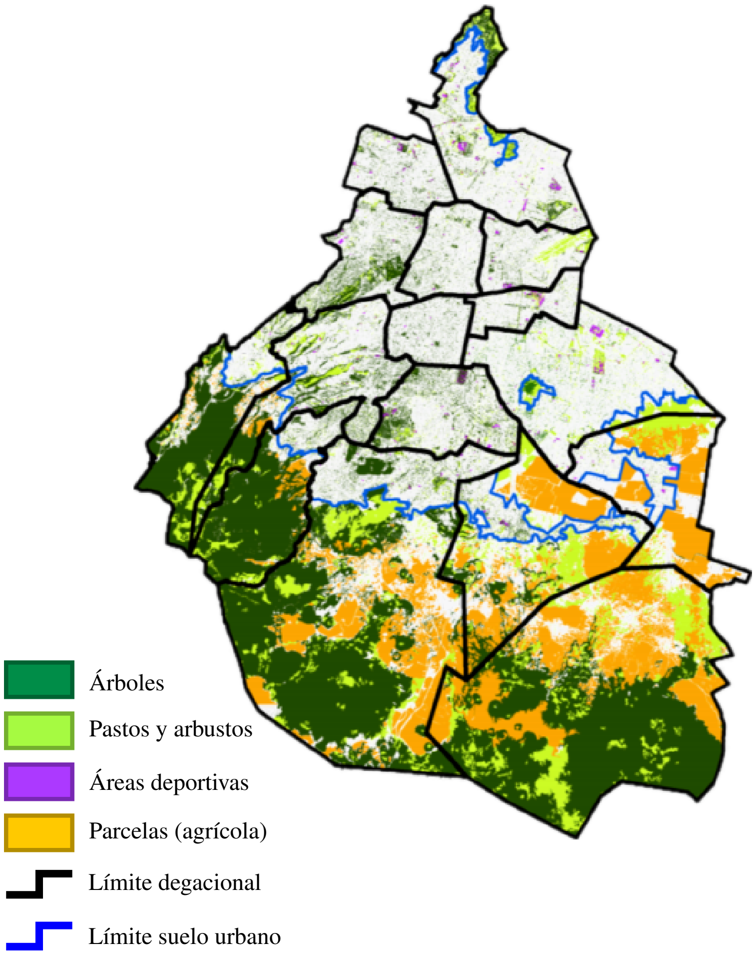


**Fig. 1.** Inventory of urban green areas in Mexico City, estimated total area of 128 km$^2$. Source PAOT 2002.

Short-term exposure to these particles is also related [24] to severe or mild cardiovascular diseases like arrhythmia, heart attack, and thrombosis. In the meanwhile, large-time exposure is associated with arterosclerosis [29] and significant health consequences in children [17]. City trees are a crucial component of the urban ecosystem. It is clear that residential trees bring benefits to society by improving the physical and mental health of its members, its economy [32], noise control, and in general the community well-being [1, 22].

From an environmental point of view, trees contribute to energy savings and influence the local weather [10]. However, the most important for this work is that the presence of urban trees is strongly related to the cities' air quality improvement. Since they are not only capable of removing air pollutants [33, 8] but also contribute to local temperature control [26], which lessens the adverse effects of pollutants.

**Table 1.** Percentage of urban trees by county, the data was obtained from the Environmental Attorney's Office and the Territorial Planning of México City.

| Counties Key | Counties | Surface area | Percentage of surface |
|---|---|---|---|
| 1 | Álvaro Obregón | 80.94 km$^2$ | 28% |
| 2 | Azcapotzalco | 33.57 km$^2$ | 11% |
| 3 | Benito Juárez | 26.77 km$^2$ | 11% |
| 4 | Coyoacán | 54.02 km$^2$ | 21% |
| 5 | Cuajimalpa de Morelos | 74.55 km$^2$ | 57% |
| 6 | Cuauhtémoc | 32.49 km$^2$ | 10% |
| 7 | Gustavo A. Madero | 87.78 km$^2$ | 11% |
| 8 | Iztacalco | 23.08 km$^2$ | 8% |
| 9 | Iztapalapa | 113.25 km$^2$ | 5% |
| 10 | Magdalena Contreras | 75.57 km$^2$ | 68% |
| 11 | Miguel Hidalgo | 46.99 km$^2$ | 26% |
| 12 | Milpa Alta | 282.72 km$^2$ | 48% |
| 13 | Tláhuac | 85.65 km$^2$ | 1% |
| 14 | Tlalpan | 307.84 km$^2$ | 45% |
| 15 | Venustiano Carranza | 33.89 km$^2$ | 8% |
| 16 | Xochimilco | 126.56 km$^2$ | 12% |

Nowadays, in Mexico City, the highest density of trees is located in particular reservoir areas [18]; they represent the 41.5% of the total green area of Mexico City this later is approximately 617.7 km$^2$, see Figure 1. The total area occupied by each county and the corresponding percentage of green areas are mentioned in Table 1. According to PAOT [1], trees, grass, and shrubs form such green areas.

Some counties have a higher percentage of green areas in the form of natural reservoirs like Magdalena Contreras. In this work, we propose a mathematical model to reforest the Mexico City counties to maximize air quality with limited resources. We consider five types of trees that are beneficial for our purposes, and present a way to distribute the limited resources among the considered counties.

In Section 2, we provide information related to the collected data on air quality and meteorological sensors all over the city, also a description of the data cleaning process. The proposed model is presented in Section 3, and its numerical solution is presented in Section 4. Finally, in Section 5, we state our conclusions and future work.

## 2 Preprocessing Data

Mexico City has a population of 9.3 million people within an area of 1,485 km$^2$ [13] with a density of 6,163.3 people/km$^2$. This high density produced, as a consequence, a reduction in green areas to build, instead, habitable and working, i.e. gray areas [31]; producing an increment in temperature.

[1] CDMX Environmental and Territorial Planning Attorney

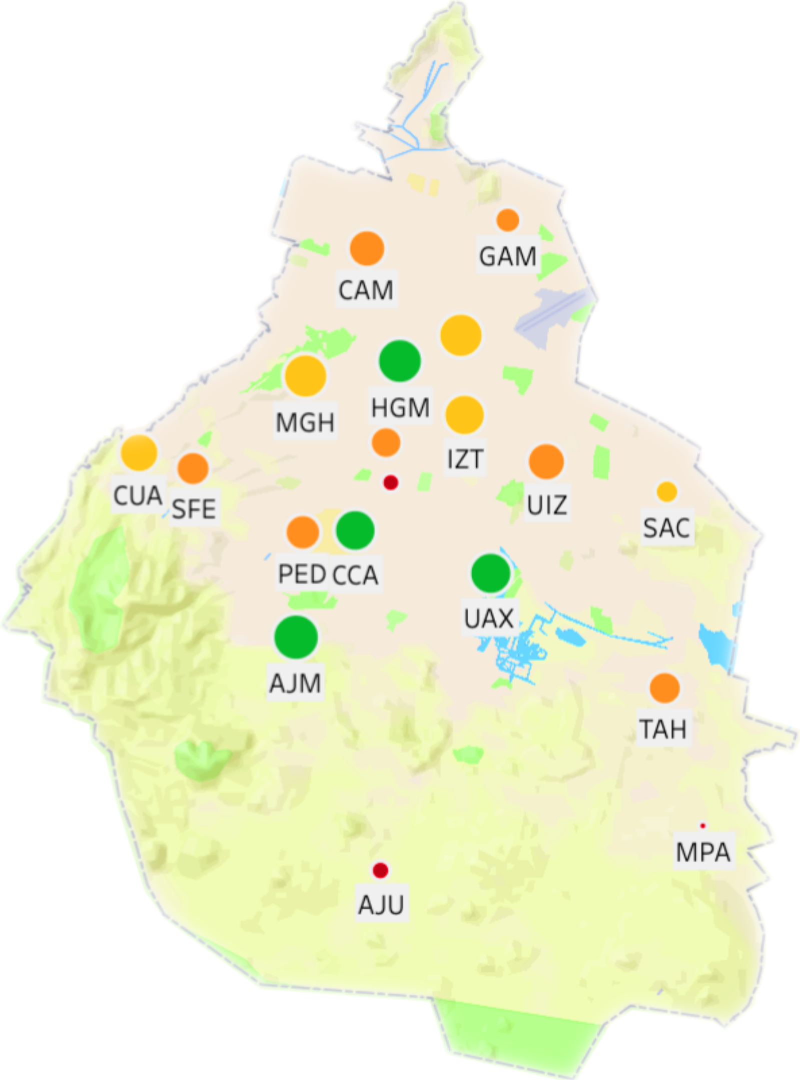


**Fig. 2.** Air quality monitoring stations located in several Mexico city's counties. The monitoring stations may change their location, depending on the needs of the Directorate of Atmospheric Monitoring of Mexico City.

The high temperatures, lack of wind, and reduced amount of surface water in the city have raised the concentration levels of $O_3$, $PM_{10}$ and $PM_{2.5}$ [34, 29]. Given the high levels of air pollution, the government of Mexico City created the Dirección de Monitoreo Atmosférico (DMA), to alert the population of the metropolitan area about the high levels of pollution of the different levels of the different pollutants and to carry out research. The DMA has pollution monitoring stations throughout the metropolitan area of Mexico, see Figure 2.

**Table 2.** Count of null values of Relative Humidity (RH), Wind Speed (WSP), and $O_3$ from 2017 to 2019 per year of the stations located in Mexico City.

| **Years** | **Schedule** | **$O_3$** | **RH** | **WSP** |
|---|---|---|---|---|
| 2018 | 24 hrs | 22023 | 14448 | 12423 |
| | 6 to 21 hrs | 12902 | 9675 | 8327 |

**Table 3.** Values of the coefficients of each variable, with standard error and the level of significance of each variable.

| **Parameters** | **coef** | **std err** | $t$ | $P > \|t\|$ |
|---|---|---|---|---|
| Interceptor | 74.6139 | 5.854 | 12.746 | 0.000 |
| $\sqrt{O_3}$ | 2.3755 | 0.594 | 3.998 | 0.000 |
| $\mathrm{WSP}^2$ | -2.1288 | 0.423 | -5.027 | 0.000 |
| $\sqrt{\mathrm{RH}/100}$ | -67.8621 | 4.905 | -13.835 | 0.000 |

Each monitoring station records the pollutant concentration levels each hour to alert the population about a possible air quality emergency. The definition for the pollution index is according to the Mexican Standard [6] of 2014. The number of active stations at year 2018 in the 16 counties from Mexico City [5], there're existence of monitoring stations may change every year; it depends on their performance during the previous year. The Atmospheric Monitoring Directorate makes the pollution and meteorological records available to the public.

These records are collected by year, they can be downloaded in various data formats, containing the pollution concentrations that each monitoring station reports every hour. A null value is placed whenever a data validation can not be carried out [15]. We noticed that some stations have certain amount of missing data[2]. Therefore, for this work, we included only 2018.

Table 2 reports the number of null values per year and separated by time. We selected no more than one station for each county. One of the discoveries made in the analysis was that the large number of null values occur from 23:00 to 5:00 hrs, so we will consider from 6:00 to 21:00 hrs. This may be due to stations maintenance it is regularly scheduled during the night.

It was found that the number of null values separated by hour is similar, which indicates that all the monitoring stations do not have a value for at least one day. For our experiments, we use the monitoring stations that have less than 10% null values. Chosen stations for analysis are those with meteorological and pollution data available. Three techniques were used to fill in the null values:

- Bootstrap method.
- Duplicate data.
- Correlation between stations.

Each method was applied depending on the number of null values for each season and month.

[2] It comes from different reasons (e.g., equipment failures or maintenance)

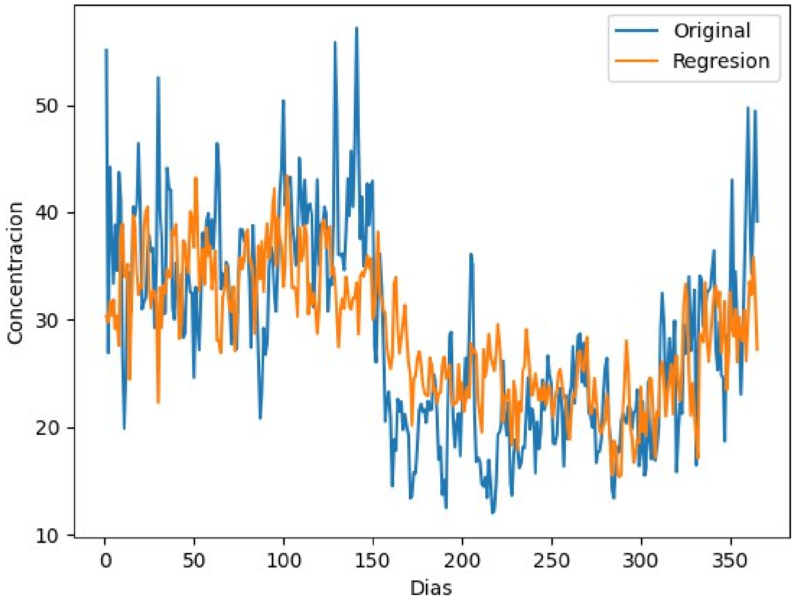


**Fig. 3.** Comparison of real data from $PM_{10}$ over the year 2018 in the AJM monitoring station against the values obtained by the model 1.

## 3 Proposed Optimization Model

Now, we introduce our model to pursue the reduction of the pollution in Mexico City, contemplating a set of 20 native trees as show list 3, the norm [28] along with the characteristics of each tree. The proposed vegetation was provided by the CONAFOR[3] and government of Mexico City. These trees have a great capacity to absorb pollution concentration.

It is worth noticing that our model considers the natural death of a certain number of trees over time after they are seeded. These tree species already exist in Mexico City, proving to be part of the existing ecosystems—this later is a very important consideration [28]. The considered species are:

- Ahuejote,
- Aile,
- Cazahuate,
- Colorin,
- Copal,
- Cuajiote,
- Encino blanco,
- Encino chico,

[3] National Forestry Commission of CDMX

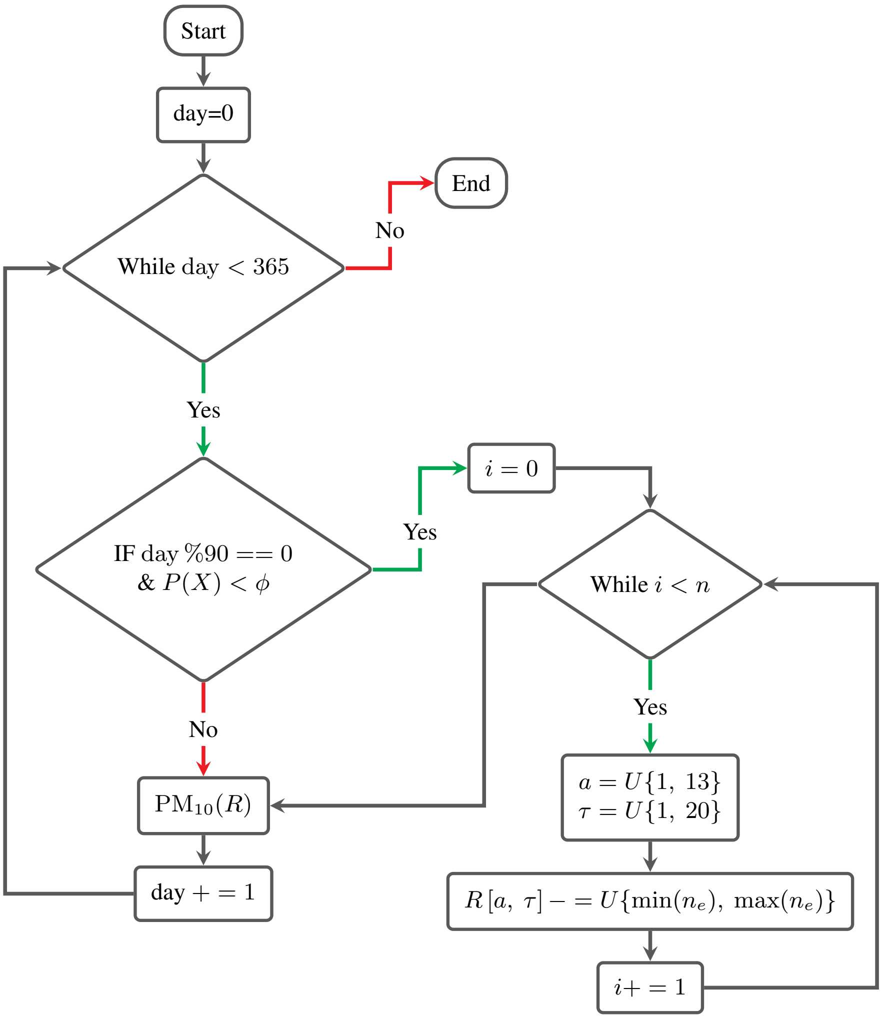
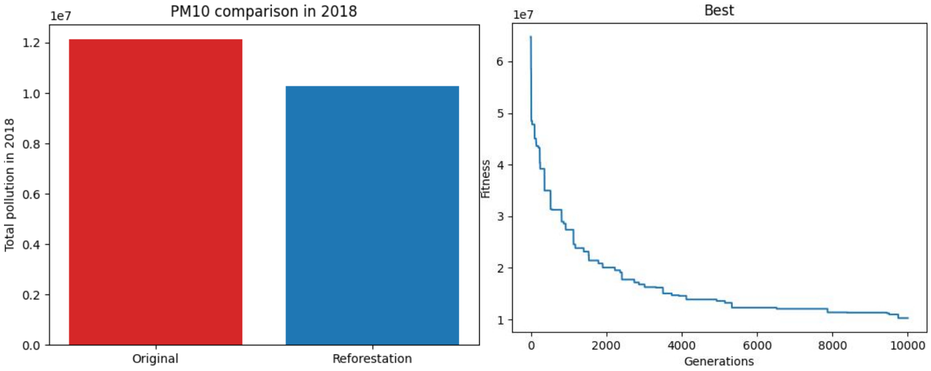


**Fig. 4.** Diagram to adjust the tree-mortality factor.

- Encino cucharita,
- Encino laurelillo,
- Encino tezahuatl,
- Jaboncillo,
- Laurel,
- Madronio,
- Mora,
- Ocote blanco,
- Palo dulce,



(a) (b)

**Fig. 5.** Comparison of the total amount of pollution in 2018 without reforestation vs with reforestation (a). Decrease in concentration of $PM_{10}$ of the individual with the best fitness across generations (b).

– Pino blanco,

– Pino patula,

– Tepozan.

In addition, we do not consider that the trees will be planted on a specific date since the government of Mexico City mentions that when carrying out a reforestation they must contemplate maintenance and continuous irrigation [5]—this depends on each species. And as [10] mentions, trees have the ability to modify the micro-climate of the area where they are found, but in our case this variation does not affect the proposed model, and therefore we can consider that the meteorological conditions are kept constant.

It is known [33, 10, 21, 22] that the variation of meteorological units like Relative humidity (RH), Wind Speed (WSP), and $O_3$, have a correlation with $PM_{10}$ concentration. Considering the Pearson correlation, we ran some tests over RH and WSP in our data set. They resulted in a negative relation, which implies that if we increase the amount of these parameters, the concentration of $PM_{10}$ will decrease. Not all the stations resulted in having the same correlation; this depends essentially on the station location.

Similarly, the relation between $O_3$ and $PM_{10}$ was verified, and we obtained that they are directly related. The $O_3$ is composed of nitrogen oxides and volatile organic compounds, the latter make up a certain proportion of $PM_{10}$ [25]. After some analysis, we propose a regression model to produce the concentrations of $PM_{10}$, under the following considerations:

1. The trees will be planted at the age of one and a half year old. They come from a nursery. This is a suitable strategy for reforestation [2].
2. The absorption percentages and contribution to levels of $O_3$, RH and WSP for each tree are known; also the absorption percentages and contribution to levels of $O_3$, RH and WSP for each tree [18].

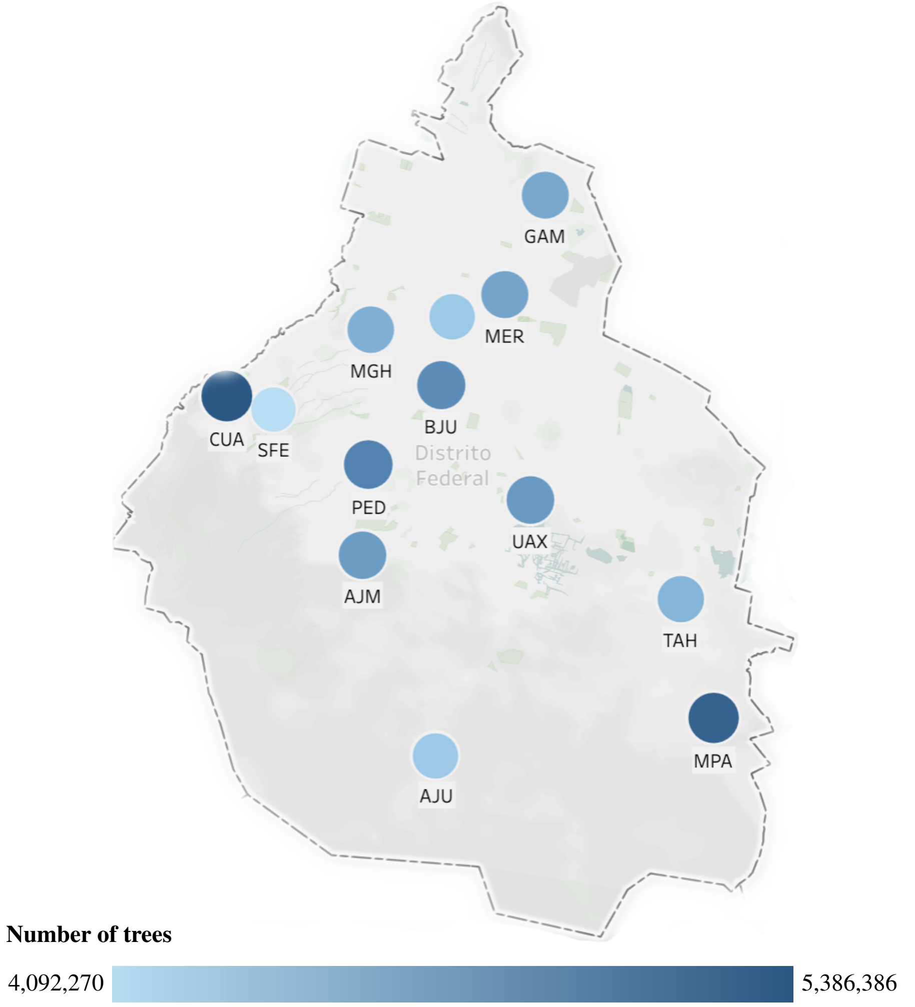


**Fig. 6.** This graph shows a color intensity related to the number of recommended trees to seed.

The proposed model to obtain the $PM_{10}$ concentrations, considering the RH, WSP and $O_3$ levels, is given in 1. Table 3 presents the statistical results and the obtained values for each coefficient:

$$\mathrm{PM}_{10} = \beta_0 + \beta_1\sqrt{O_3} + \beta_2\mathrm{WSP}^2 + \beta_3\sqrt{\frac{\mathrm{RH}}{100}}. \tag{1}$$

It was shown that the $PM_{10}$ function is a good approximation to the real values and the Weierstrass test was performed to determine that both distributions were similar.At the end it was obtained that both distributions are similar with a p-value greater than 0.05. The figure 3 shows the $PM_{10}$ function in orange and the real values in blue.

**Table 4.** Values of the parameters used with GA.

| Parameter | Value |
|---|---|
| Population | 100 |
| Cross | 0.9 |
| Mutation | 0.1 |
| Mortality | 0.1 |
| Generations | 10000 |

We computed the $R^2$ coefficient to verify the proposed model's relation against past-year measured values records. The proposal is related in a ratio of 42.5% to the real values. Finally, due to the validation process, we confirmed that the proposal does not have any dependent variable, so our model fulfills the independence assumption.

Each variable was analyzed, in periods of one hour, separately to see its effect on $\text{PM}_{10}$. Table 3 shows that each variable is significant for our proposal. Figure 3 presents the comparison for the regression against real observed values. Carrying out a T-Student test with a null hypothesis sample that both means are statistically equal.

The hypothesis will be accepted with a significance level greater than or equal to 5% [27]. The goal is to minimize the amount of $\text{PM}_{10}$ concentration in the air over one entire 1 year (365 days) by reforesting using a particular distribution of trees over the sixteen several counties of the city. The objective function that we considered in this work is presented in (2):

$$\min_{\text{PM}_{10}} \; f_1(\delta) = \sum_{i=1}^{16} \left( \sum_{j=1}^{365} \text{PM}_{10}(\delta) \right), \tag{2}$$

where:

$$\text{PM}_{10} = \beta_0 + [T] \left( \%\text{O}_3 \sqrt{\text{O}_3} + \%\text{WSP}\;\beta_2\;\text{WSP}^2 + \%\text{RH}\;\beta_3\;\sqrt{\frac{\text{RH}}{100}} \right).$$

And $\delta = [[T]_i,\ \text{O}_{3i},\ \text{WSP}_i,\ \text{RH}_i]$. Each tree has an absorption percentage of $\text{O}_3$ and a percentage increase of RH y WSP. We should mention that our model considers the amount of pollen released by each one of the seeded trees, that also contribute to a raise in $\text{PM}_{10}$ levels; otherwise, the problem could have a trivial solution. It is worth to notice that (2) implies a discrete problem.

### 3.1 Genetic Algorithm

**Representation.** The representation of an individual is a random $R_{13\times 20}$ matrix for each $T_{a,\tau}$ is a mayor and a type of tree 3:

$$R_i = \begin{bmatrix} T_{1,1} & T_{1,2} & \dots & T_{1,20} \\ T_{2,1} & T_{2,2} & \dots & T_{2,20} \\ \vdots & \vdots & \ddots & \vdots \\ T_{13,1} & T_{13,2} & \dots & T_{13,20} \end{bmatrix}. \tag{3}$$

**GA operators:**

- **Mutation**: A tree is taken at random with probability $\eta$ and is varied between 0 and 5000000.
- **Selection**: We take 50 individuals with the highest fitness using a tournament with 3 individuals and no replacement.
- **Crossover**: A pair of parents is taken at random with a probability of $\mu$ with a one-point crossover by columns to generate two offspring that will replace them.
- **Fitness Evaluation**: In our problem, we define the fitness of an individual as the amount of PM$_{10}$ in 365 days.

**Tree-mortality factor.** As we know when doing a reforestation there is a probability that some of the trees that are planted will die either by a pest, among others [7]. The operator is used when evaluating the individual in $PM_{10}$ function. It randomly chooses a certain number of trees to remove from $R$ every 90 days with a probability of $\phi$.

To select a $T_{a,\tau}$, $n$ elements are taken from $R$ with the same probability, and to determine the number of trees to be removed from $R$ with the same probability, $T_{a,\tau}$ number of trees to be removed we take a random number between minimum and maximum of $T_{a,\tau}$ from the $n$ elements selected, the $n$ selected elements. Figure 4 shows the steps followed to perform this process.

**Evaluation.** To do the evaluation of $PM_{10}$ we use $R$ as well as the database of the atmospheric monitoring address that is preprocessed.

## 4 Problem-Solving and Results

The coding for solving the problem was developed in Python to obtain the best individual of all the generations. The following conditions were considered for our experiments:

- data is taken from 6 to 22 hrs
- stations are located within the CDMX, and
- the data is taken as the average value per day.

To obtain the values of each parameter shown in the table 4, we executed several runs varying the probabilities $\mu$, $\eta$ and $\phi$ and then, observed the following:

- If $\phi > 0.1$ then in less than 100 generations the amount of available trees was zero.
- Using the one-point crossover yielded better results than the two-point crossover because the problem is sensitive to large changes.
- If $\eta > 0.1$ it was observed that the algorithm convergence was severely affected.

The result presented was the best of 32 independent experiments with the same, best found, parameters. Although a county may have more than one monitoring station, only one station was considered per county.

**Table 5.** Optimal tree distribution by county.

| | $\tau_1$ | $\tau_2$ | $\tau_3$ | $\tau_4$ | $\tau_5$ | $\tau_6$ | $\tau_7$ | $\tau_8$ | $\tau_9$ | $\tau_{10}$ |
|---|---|---|---|---|---|---|---|---|---|---|
| $A_1$ | 418309 | 375719 | 303090 | 94599 | 334681 | 217567 | 195576 | 228510 | 7667 | 398063 |
| $A_2$ | 144408 | 3845 | 470600 | 302911 | 190923 | 60659 | 399995 | 189185 | 276041 | 54518 |
| $A_3$ | 50033 | 124994 | 207623 | 43972 | 58405 | 224505 | 296112 | 461770 | 209919 | 98022 |
| $A_4$ | 108242 | 332672 | 414557 | 141839 | 35557 | 296881 | 44946 | 397703 | 228142 | 57399 |
| $A_5$ | 80327 | 329660 | 349215 | 451593 | 27456 | 341509 | 469595 | 307736 | 313247 | 238416 |
| $A_6$ | 116992 | 277891 | 436061 | 220096 | 264982 | 333060 | 304585 | 393349 | 69190 | 175873 |
| $A_7$ | 265002 | 176476 | 163733 | 208879 | 309005 | 154325 | 168721 | 336517 | 404519 | 350366 |
| $A_8$ | 170870 | 165178 | 56430 | 497019 | 190832 | 236261 | 134427 | 484378 | 86706 | 427950 |
| $A_9$ | 350186 | 406475 | 316707 | 309593 | 412799 | 432951 | 21011 | 262130 | 274982 | 38417 |
| $A_{10}$ | 138705 | 188230 | 492681 | 326405 | 71313 | 50313 | 179803 | 291370 | 64 | 286392 |
| $A_{11}$ | 16093 | 308462 | 140133 | 260273 | 481663 | 39364 | 294260 | 306817 | 394137 | 122410 |
| $A_{12}$ | 62941 | 498795 | 250330 | 250821 | 8510 | 40430 | 318980 | 446872 | 74679 | 289328 |
| $A_{13}$ | 404245 | 128481 | 37746 | 390207 | 158925 | 7715 | 119427 | 499842 | 83427 | 46076 |

| | $\tau_{11}$ | $\tau_{12}$ | $\tau_{13}$ | $\tau_{14}$ | $\tau_{15}$ | $\tau_{16}$ | $\tau_{17}$ | $\tau_{18}$ | $\tau_{19}$ | $\tau_{20}$ |
|---|---|---|---|---|---|---|---|---|---|---|
| $A_1$ | 148654 | 236331 | 464689 | 105415 | 421296 | 134009 | 194197 | 26216 | 146588 | 285046 |
| $A_2$ | 207203 | 119750 | 19095 | 328261 | 323703 | 12957 | 468505 | 135282 | 325336 | 59093 |
| $A_3$ | 289234 | 125286 | 102826 | 259699 | 280368 | 494208 | 342552 | 219431 | 462057 | 126735 |
| $A_4$ | 213177 | 398086 | 472656 | 64368 | 403191 | 279417 | 154335 | 308697 | 81133 | 277557 |
| $A_5$ | 211402 | 264516 | 80493 | 252369 | 332070 | 448782 | 143707 | 284814 | 11522 | 348147 |
| $A_6$ | 359841 | 448520 | 394401 | 425817 | 765 | 12978 | 312141 | 42890 | 241506 | 15142 |
| $A_7$ | 312749 | 379449 | 273885 | 9708 | 176994 | 157576 | 152870 | 268122 | 148644 | 182370 |
| $A_8$ | 264080 | 45272 | 226295 | 261476 | 324569 | 170519 | 83320 | 258113 | 251907 | 190088 |
| $A_9$ | 128023 | 172077 | 466581 | 215939 | 81869 | 430903 | 41311 | 195731 | 356574 | 472127 |
| $A_{10}$ | 480242 | 188932 | 441634 | 166075 | 48421 | 283771 | 210312 | 165914 | 247269 | 33441 |
| $A_{11}$ | 324561 | 196006 | 286242 | 397527 | 89466 | 179092 | 309688 | 303896 | 29336 | 151566 |
| $A_{12}$ | 291557 | 85977 | 247820 | 282415 | 40763 | 298595 | 469195 | 363578 | 312774 | 305990 |
| $A_{13}$ | 8642 | 69473 | 219038 | 265509 | 39713 | 340457 | 151027 | 416879 | 408536 | 499906 |

The results obtained from our experiments are promising because it suggests that by making an optimal distribution of the locations where each tree is to be planted, better air quality can be obtained. An important thing to consider is that the algorithm does not mention the exact coordinates where each type of tree should be planted inside the county.

The individual with the best suitability represents 60,819,340 trees, and decreases pollution by 15% with respect to data obtained from the 2018 CDMX Atmospheric Monitoring Directorate. Figure 5 shows the sum of $PM_{10}$ concentration in 2018 without using our model and blue color shows the sum of $PM_{10}$ concentration of 2018 using our model. These results are in good agreement with the results obtained in [23, 30].

Figure 5 shows the fitness of the best individual in all generations and what is observed is that in the first 5000 generations it decreases rapidly and after 5000 generations the speed with which the amount of $\mathrm{PM}_{10}$ concentration decreases. And although it stops in the 10000th generation, our tree distribution improved the air quality. Figure 6 illustrates the number of trees obtained according to each considered municipality.

The dots do not indicate the precise coordinate where they will be planted. The total number of trees varies from 4 million to 5 million approximately. Table 5 shows the obtained optimal distribution of native trees in the thirteen municipalities of Mexico City. The maximum number of trees of any specie to be planted is 499906, and it corresponds to the municipality $A_{13}$ with the $\tau_{20}$ type of tree, while the minimum number of trees is 64 and corresponds to municipality $A_{10}$ and the $\tau_9$ specie.

## 5 Conclusions

In this work, we introduced a model to improve air quality in Mexico City by urban reforestation. We considered a constrained discrete optimization problem and used a genetic algorithm to obtain an optimal distribution of trees over the different counties. We used a prediction model for the solution after publicly available data about air quality and meteorological values measured along the city.

Assuming the percentages of RH, WSP and $\mathrm{O}_3$ absorption for twenty considered types of trees, we were able to reduce the amount of total $\mathrm{PM}_{10}$ concentration. An important aspect to notice was the necessary data cleaning process since efficiently filling the database was essential for the model's success.

In 2018 Mexico City experienced an increase in the amount of pollution with an amount of $12.1 \times 10^6$ $\mu\mathrm{g/m}^3$ of $\mathrm{PM}_{10}$, and the solution obtained by simulation, after apply this proposal was $10.2 \times 10^6$; this implies at least a 15% possible improvement. In addition, our model uses native trees from Mexico City, meaning a better resistance to the local climate and fewer tree deaths.

As a future work, the problem will be approached as a multi-objective problem, with the cost of planting and maintenance of the trees as a second function. Also, the characteristics of the trees will be added as another objective function to determine the mortality of the trees using clustering techniques. And to predict the reforestation behavior along 10 years.

**Acknowledgements.** Adriana Lara acknowledges project IPN-SIP 20231043.

## References

1. Akbari, H., Taha, H.: The impact of trees and white surfaces on residential heating and cooling energy use in four canadian cities. Energy, vol. 17, no. 2, pp. 141–149 (1992) doi: 10.1016/0360-5442(92)90063-6
2. Arriaga, V., Cervantes, V., Vargas-Mena, A.: Manual de reforestación con especies nativas. Secretaría de Desarrollo Social. Instituto Nacional de Ecología (1994)

3. Bernstein, J. A., Alexis, N., Barnes, C., Bernstein, I. L., Nel, A., Peden, D., Diaz-Sanchez, D., Tarlo, S. M., Williams, P. B., Bernstein, J. A.: Health effects of air pollution. Journal of Allergy and Clinical Immunology, vol. 114, no. 5, pp. 1116–1123 (2004) doi: 10.1016/j.jaci.2004.08.030
4. Canales-Rodríguez, M. Á., Quintero-Núñez, M., Castro-Romero, T. G., García-Cuento, R. O.: Las partículas respirables $PM_{10}$ y su composición química en la zona urbana y rural de Mexicali, Baja California en México. Información tecnológica, vol. 25, no. 6, pp. 13–22 (2014) doi: 10.4067/s0718-07642014000600003
5. Ciudad de México, Administración Pública: Gaceta oficial de la Ciudad de México, vol. 498, pp. 12–13 (2018)
6. Diario Oficial de la Federación: Normas oficiales mexicanas (2014) www.dof.gob.mx
7. Elster, C.: Reasons for reforestation success and failure with three mangrove species in colombia. Forest Ecology and Management, vol. 131, no. 1-3, pp. 201–214 (2000) doi: 10.1016/s0378-1127(99)00214-5
8. Escobedo, F. J., Kroeger, T., Wagner, J. E.: Urban forests and pollution mitigation: Analyzing ecosystem services and disservices. Environmental Pollution, vol. 159, no. 8-9, pp. 2078–2087 (2011) doi: 10.1016/j.envpol.2011.01.010
9. García-de Jalón, S., Burgess, P. J., Curiel-Yuste, J., Moreno, G., Graves, A., Palma, J. H. N., Crous-Duran, J., Kay, S., Chiabai, A.: Dry deposition of air pollutants on trees at regional scale: A case study in the basque country. Agricultural and Forest Meteorology, vol. 278, pp. 107648 (2019) doi: 10.1016/j.agrformet.2019.107648
10. Georgi, N. J., Zafiriadis, K.: The impact of park trees on microclimate in urban areas. Urban Ecosystems, vol. 9, no. 3, pp. 195–209 (2006) doi: 10.1007/s11252-006-8590-9
11. Hanna, R., Oliva, P.: The effect of pollution on labor supply: Evidence from a natural experiment in Mexico City. Journal of Public Economics, vol. 122, pp. 68–79 (2015) doi: 10.1016/j.jpubeco.2014.10.004
12. Héroux, M.-E., Anderson, H. R., Atkinson, R., Brunekreef, B., Cohen, A., Forastiere, F., Hurley, F., Katsouyanni, K., Krewski, D., Krzyzanowski, M., Künzli, N., Mills, I., Querol, X., Ostro, B., Walton, H.: Quantifying the health impacts of ambient air pollutants: Recommendations of a WHO/Europe project. International Journal of Public Health, vol. 60, no. 5, pp. 619–627 (2015) doi: 10.1007/s00038-015-0690-y
13. Instituto Nacional de Estadística y Geografía: Censo población y vivienda (2021) www.inegi.org.mx/programas/ccpv/2020/default.html
14. Janssen, N. A. H., Hoek, G., Simic-Lawson, M., Fischer, P., van Bree, L., ten Brink, H., Keuken, M., Atkinson, R. W., Anderson, H. R., Brunekreef, B., Cassee, F. R.: Black carbon as an additional indicator of the adverse health effects of airborne particles compared with $PM_{10}$ and $PM_{2.5}$. Environmental Health Perspectives, vol. 119, no. 12, pp. 1691–1699 (2011) doi: 10.1289/ehp.1003369
15. Kruschke, J. K.: Bayesian assessment of null values via parameter estimation and model comparison. Perspectives on Psychological Science, vol. 6, no. 3, pp. 299–312 (2011) doi: 10.1177/1745691611406925
16. Le, H. D., Smith, C., Herbohn, J., Harrison, S.: More than just trees: Assessing reforestation success in tropical developing countries. Journal of Rural Studies, vol. 28, no. 1, pp. 5–19 (2012) doi: 10.1016/j.jrurstud.2011.07.006
17. Loomis, D., Castillejos, M., Gold, D. R., McDonnell, W., Borja-Aburto, V. H.: Air pollution and infant mortality in Mexico City. Epidemiology, vol. 10, no. 2, pp. 118–123 (1999)
18. López-López, S. F., Martínez-Trinidad, T., Benavides-Meza, H. M., García-Nieto, M., Ángeles Pérez, G.: Reservorios de biomasa y carbono en el arbolado de la primera sección del Bosque de Chapultepec, Ciudad de México. Madera y Bosques, vol. 24, no. 3 (2018) doi: 10.21829/myb.2018.2431620

19. Mayer, H.: Air pollution in cities. Atmospheric Environment, vol. 33, no. 24-25, pp. 4029–4037 (1999) doi: 10.1016/s1352-2310(99)00144-2
20. Molina, R. T., Rodríguez, A. M., Palaciso, I. S., López, F. G.: Pollen production in anemophilous trees. Grana, vol. 35, no. 1, pp. 38–46 (1996) doi: 10.1080/00173139609430499
21. Nowak, D.: Air pollution removal by Chicago's urban forest. Chicago's Urban Forest Ecosystem: Results of the Chicago Urban Forest Climate Project, pp. 63–81 (1994)
22. Nowak, D. J., Crane, D. E., Stevens, J. C.: Air pollution removal by urban trees and shrubs in the United States. Urban forestry and urban greening, vol. 4, no. 3-4, pp. 115–123 (2006)
23. Nowak, D. J., Hirabayashi, S., Bodine, A., Hoehn, R.: Modeled $PM_{2.5}$ removal by trees in ten U.S. cities and associated health effects. Environmental Pollution, vol. 178, pp. 395–402 (2013) doi: 10.1016/j.envpol.2013.03.050
24. Pope, C. A., Ezzati, M., Dockery, D. W.: Fine-particulate air pollution and life expectancy in the United States. New England Journal of Medicine, vol. 360, no. 4, pp. 376–386 (2009) doi: 10.1056/nejmsa0805646
25. Querol, X.: $PM_{10}$ and $PM_{2.5}$ source apportionment in the Barcelona Metropolitan area, Catalonia, Spain. Atmospheric Environment, vol. 35, no. 36, pp. 6407–6419 (2001) doi: 10.1016/s1352-2310(01)00361-2
26. Ries, K., Eichhorn, J.: Simulation of effects of vegetation on the dispersion of pollutants in street canyons. Meteorologische Zeitschrift, vol. 10, no. 4, pp. 229–233 (2001) doi: 10.1127/0941-2948/2001/0010-0229
27. Ruxton, G. D.: The unequal variance t-test is an underused alternative to student's t-test and the Mann–Whitney U test. Behavioral Ecology, vol. 17, no. 4, pp. 688–690 (2006) doi: 10.1093/beheco/ark016
28. Schaberg, P. G., DeHayes, D. H., Hawley, G. J., Nijensohn, S. E.: Anthropogenic alterations of genetic diversity within tree populations: Implications for forest ecosystem resilience. Forest Ecology and Management, vol. 256, no. 5, pp. 855–862 (2008) doi: 10.1016/j.foreco.2008.06.038
29. Shields, K. N., Cavallari, J. M., Hunt, M. J. O., Lazo, M., Molina, M., Molina, L., Holguin, F.: Traffic-related air pollution exposures and changes in heart rate variability in Mexico City: A panel study. Environmental Health, vol. 12, no. 1 (2013) doi: 10.1186/1476-069x-12-7
30. Soares, A. L., Rego, F. C., McPherson, E. G., Simpson, J. R., Peper, P. J., Xiao, Q.: Benefits and costs of street trees in Lisbon, Portugal. Urban Forestry and Urban Greening, vol. 10, no. 2, pp. 69–78 (2011) doi: 10.1016/j.ufug.2010.12.001
31. Sobrino, J.: La urbanización en el méxico contemporáneo. Centro Latinoamericano y Caribeño de Demografía Comisión Económica para América Latina y el Caribe, (2012)
32. Soledad-Duval, V., Graciela-Benedetti, M., Baudis, K.: El impacto del arbolado de alineación en el microclima urbano. bahía blanca, argentina. Investigaciones Geográficas, vol. 1, no. 73, pp. 171–188 (2020)
33. Yang, J., McBride, J., Zhou, J., Sun, Z.: The urban forest in beijing and its role in air pollution reduction. Urban Forestry and Urban Greening, vol. 3, no. 2, pp. 65–78 (2005) doi: 10.1016/j.ufug.2004.09.001
34. Zuk, M.: Tercer almanaque de datos y tendencias de la calidad del aire en nueve ciudades mexicanas. Instituto Nacional de Ecología (2007)

ISSN 1870-4069

# Development of a Front-End with Dynamic Searches for Chatbot Retraining Using ReactJS

Marco A. Bosquez-González[1], Luis E. Bojórquez-Almazán[1],
Kevin G. Zazueta-Sánchez[2], Yoel Ledo-Mezquita[1],
Elioth Macias-Frotto[1]

[1] Tecnologico de Monterrey,
School of Engineering and Sciences,
Mexico

[2] NDS Cognitive Labs.,
Mexico

{a01653247, a01336625, yledo, a00837025}@tec.mx,
kzazueta@ndscognitivelabs.com

**Abstract.** This article will show the development of a web platform using ReactJS for dynamic document searches, and the collaboration of a front-end team in the project "Virtual Assistant Management Platform" for Chatbot Retraining. Copyright © 2022 Tec de Monterrey and NDS Cognitive Labs.

**Keywords:** Chatbots, dynamic search, front-end, reactJS, computing.

## 1 Introduction

In order to give a real work experience to Tecnológico de Monterrey students, calls were opened for stays or professional internships that would begin for the semester of the period August – December 2022. One of the companies that would be part of this experience is NDS Cognitive Labs, which is located in Mexico and the United States.

NDS Cognitive Labs is responsible for developing cognitive computing and innovating technology in Mexico. To this end, it was decided to enrich the profile of the students by participating in professional experiences and developing real skills [1].

In conjunction with Tecnológico de Monterrey and NDS Cognitive Labs, these calls were opened to Tecnológico de Monterrey students. At the opening of the interviews, the professor and person in charge of communication between NDS and Tecnológico de Monterrey, Dr. Yoel Ledo, together with the coordinator of the professional stay on behalf of NDS Cognitive Labs, Dr. Raúl Morales, presented the different experiences.

These experiences were assigned based on the student's profile, in this way the most suitable one for the members could be defined. For this article, experience 3, "Virtual Assistant Management Platform", will be highlighted. This experience is described as the development of a Natural Language Processing (NLP) platform, which would use tools such as ReactJS, Javascript, Python, MongoDB, Google Cloud Platform, Flask and IBM Watson.

As part of this experience, a chatbot was built using IBM Watson, which handles artificial intelligence (AI) and machine learning (ML) technologies [2]. The foundation

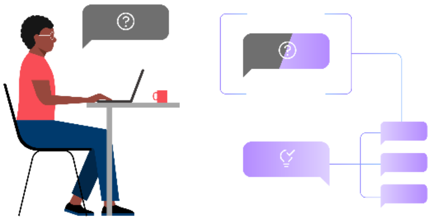


**Fig. 1.** IBM. (October, 2022). Leading conversation.

**Fig. 2.** IBM. (October, 2022). Simple construction.

of the IBM Cloud platform made it possible to create a Bot based on an advanced and publicly available machine learning system.

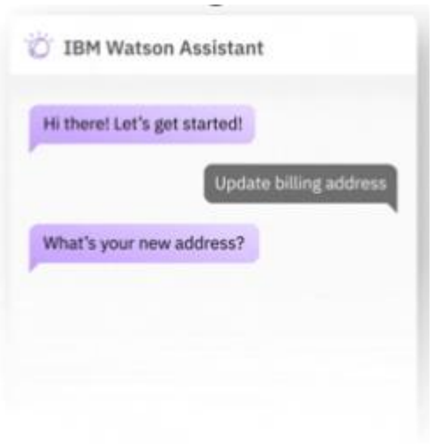
Chatbots are a useful way for customers to ask questions and get answers right away. The Bot can answer questions or direct the customer to other resources. These chatbots can run within a wide range of devices, applications, channels, and mobiles [3]. Several webpages were also used as the basis for the construction of our own model [4-6].

## 2 Materials and Methods

### 2.1 Second Level Title

Through IBM Watson that was mentioned above, the communication tool between the chatbot and the customer was created. The input of information in this case would be the client's request that would then be processed through natural language to then choose the best possible response.

For the creation of the dialogue with the chatbot, it is important to use methods that represent the user's information that is relevant to the user's purpose. These tools are known as intentions and entities [7].



Hi there! Let's get started!

Update billing address

What's your new address?

**Fig. 3.** IBM. (November, 2021). Comprehensive.

Intentions are defined as verbs and actions that the user wants to perform, while entities are nouns such as objects, places, or things related to a business. The chatbot gathers and analyzes them to get a possible answer.

### 2.2 Project Context

By using the technologies and tools mentioned above. The project of this work experience sought to develop a platform in which the creation, editing, visualization, deletion and search of intentions and entities can be managed, in order to have all the possible answers to questions that are asked.

By correctly defining the intentions and entities, the aim is to gather the relevant information to determine the purpose of the request requested by a potential client. It has the ability to assign names, assign different types of values, and bind intents and entities. With this data, the chatbot algorithm developed by IBM Watson was trained.

For this project, the MongoDB NoSQL database was used, in which documents are stored, which are data records in BSON-formatted objects. These documents are separated into collections, in which intentions and entities are separated. In this way, this database is used to perform CRUD functions on these data types.

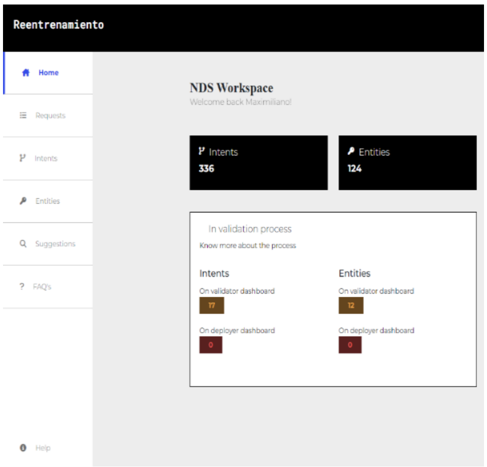
In conjunction with MongoDB and IBM Watson, we developed a platform that allows us to handle CRUD functions of intents and entities in a better way. This platform is accessible to users in a way that can be accessed from the computer or a mobile device.

All this set of technologies, methods, natural language processes and data help us to be able to retrain the chatbot with all the possible answers to all the possible questions that the user has.

### 2.3 Work Sharing

Taking into account the technologies and tools to be used, as well as the profile of the members of the experience, the work to be carried out was distributed among the first

**Fig. 4.** MongoDB. (2022). Homepage Hero. MongoDB, Inc.



NDS Workspace
Welcome back Maximiliano!

Intents
336

Entities
124

In validation process
Know more about the process

Intents
On validator dashboard
17
On deployer dashboard
0

Entities
On validator dashboard
12
On deployer dashboard
0

**Fig. 5.** (2022). Chatbot Retraining Platform.

weeks of working in the team. The work teams within this project were split between the back-end team that would work with Python, Flask, and MongoDB, and the front-end team that would work primarily with ReactJS.

Seven students from Tecnológico de Monterrey took part in this experience, so, without taking into account the leaders of the experiences, initially 5 people would be distributed in the back-end team and 2 people in the front-end team.

The front-end team to which the authors of this article mainly belong, were selected due to their previous experiences with the open-source front-end JavaScript library, ReactJS. Likewise, due to previous experiences in Python and MongoDB, you would have the opportunity to collaborate and contribute to the back-end team.

The experience leaders would give the instruction to take MongoDB courses on their official page to relate to the database and its relationship with the development of the platform.

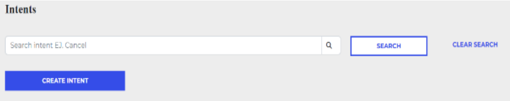
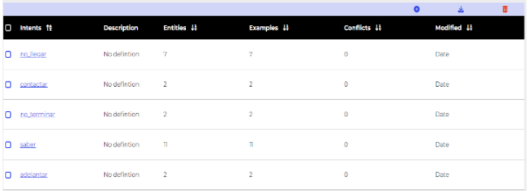
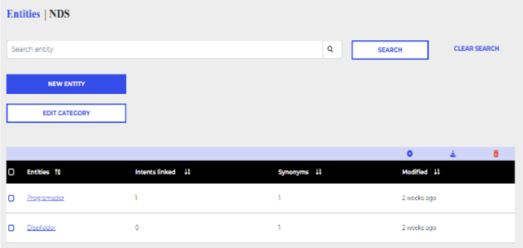


**Fig. 6.** (2022). Search Bar.



| □ | Intents | Description | Entities | Examples | Conflicts | Modified |
|---|---|---|---|---|---|---|
| □ | no_llegar | No defintion | 7 | 7 | 0 | Date |
| □ | contactar | No defintion | 2 | 2 | 0 | Date |
| □ | no_terminar | No defintion | 2 | 2 | 0 | Date |
| □ | saber | No defintion | 11 | 11 | 0 | Date |
| □ | adelantar | No defintion | 2 | 2 | 0 | Date |

**Fig. 7.** (2022). Table of Intentions.



| □ | Entities | Intents linked | Synonyms | Modified |
|---|---|---|---|---|
| □ | Programador | 1 | 1 | 2 weeks ago |
| □ | Diseñador | 0 | 1 | 2 weeks ago |

**Fig. 8.** (2022). Table of Entities and Search Bar.

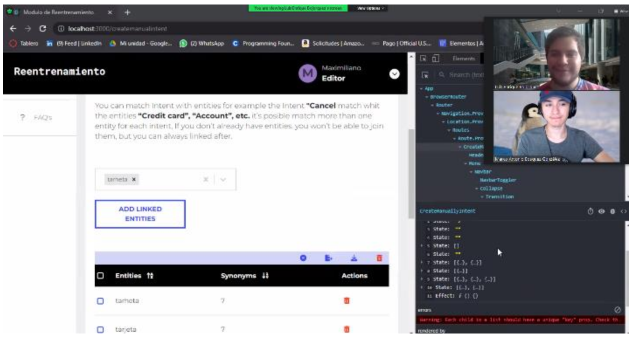
As for the way the team works, the members of the front-end team, Marco A. Bósquez González and Luis E. Bojórquez Almazán, were presented with the base platform with which they were going to work. The base of the platform has the defined components to work with and that would need to be operationalized. Likewise, it was tested to observe the flow of the platform simulating its use for end users.

The components to be worked on include working the creation, editing, viewing and deletion screens, using ReactJS. At the back of the page, endpoints developed by the back-end team would be used to extract the information and data necessary to give the platform and components a proper function.

## 2.4 Communication & Development

As for the communication and development of the project by the back-end team, every week meetings are scheduled through the Google Meets platform in which the weekly progress is shown and the functionalities to be developed for the next reviews are discussed. Additionally, in order to have a record of the progress of the project, weekly logs were written about the main points of each week and the constant progress of the application. This was demonstrated by evidence that is included in the logs.

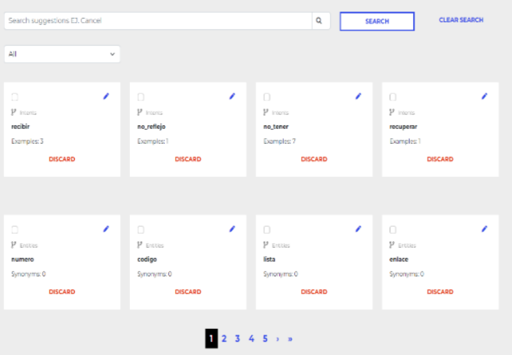
The project leader, engineer Kevin G. Zazueta Sanchez, reviewed the progress of the team consisting of Marco A. Bosquez Gonzalez and Luis E. Bojorquez Almazan. In the same way, in case of needing support from the leader, video calls are scheduled in order to resolve doubts and develop the aforementioned components. As for the way the front-end team works, the members worked in different perspectives in which the



You can match Intent with entities for example the Intent **"Cancel** match whit the entities **"Credit card", "Account", etc.** it's posible match more than one entity for each intent. If you don't already have entities you won't be able to join them, but you can always linked after.

| ☐ | Entities | Synonyms | Actions |
|---|---|---|---|
| ☐ | tarhota | 7 | 🗑 |
| ☐ | tarjeta | 7 | 🗑 |

**Fig. 9.** Marco A. Bosquez González, Luis E. Bojórquez. (2022). Zoom meeting.



| | | | |
|---|---|---|---|
| ☐ Intents<br>**recibir**<br>Examples: 3<br>DISCARD | ☐ Intents<br>**no_reflejo**<br>Examples: 1<br>DISCARD | ☐ Intents<br>**no_tener**<br>Examples: 7<br>DISCARD | ☐ Intents<br>**recuperar**<br>Examples: 1<br>DISCARD |
| ☐ Entities<br>**numero**<br>Synonyms: 0<br>DISCARD | ☐ Entities<br>**codigo**<br>Synonyms: 0<br>DISCARD | ☐ Entities<br>**lista**<br>Synonyms: 0<br>DISCARD | ☐ Entities<br>**enlace**<br>Synonyms: 0<br>DISCARD |

**Fig. 10. (2022).** Suggestions View with Search Bar and Selection of Options to Display.

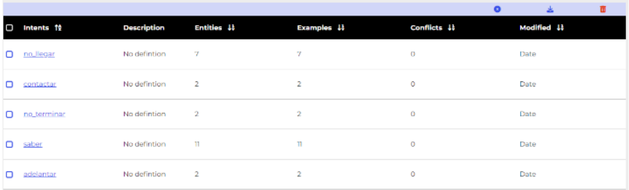
defined components were developed. This is in order to get the components up and running as quickly as possible and move on to the next components. Sometimes the support of both members is required to solve a problem, so there is constant communication in the front-end team.

The front-end project works with a repository on GitLabs in which versions are maintained that different team members can work with. The front-end team, which is composed of Marco A. Bósquez González and Luis E. Bojórquez Almazán, worked mainly on all the views that include intentions, entities, and suggestions, so they worked on a specific branch.

## 3 Results

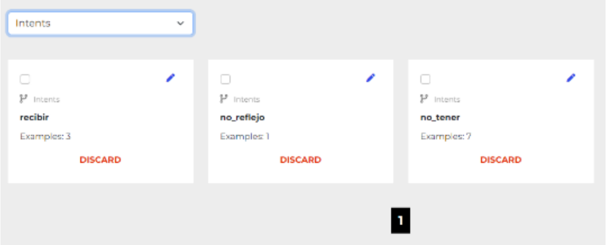
By using ReactJS and the endpoints created by the back-end team, the search bar component could be worked on and added in most of the platform's views, including intents, entities, suggestions, and requests.

The search bar makes use of inputs, useState useEffect hook functions, component rendering, Axios, and components from React's library, MaterialUI (MUI). All of this makes up the search bar and its buttons that update every time you type or perform a search.



| □ | Intents ⇅ | Description | Entities ⇅ | Examples ⇅ | Conflicts ⇅ | Modified ⇅ |
|---|---|---|---|---|---|---|
| □ | no_llegar | No defintion | 7 | 7 | 0 | Date |
| □ | contactar | No defintion | 2 | 2 | 0 | Date |
| □ | no_terminar | No defintion | 2 | 2 | 0 | Date |
| □ | saber | No defintion | 11 | 11 | 0 | Date |
| □ | adelantar | No defintion | 2 | 2 | 0 | Date |

**Fig. 11.** (2022). Table of Intentions.



Intents

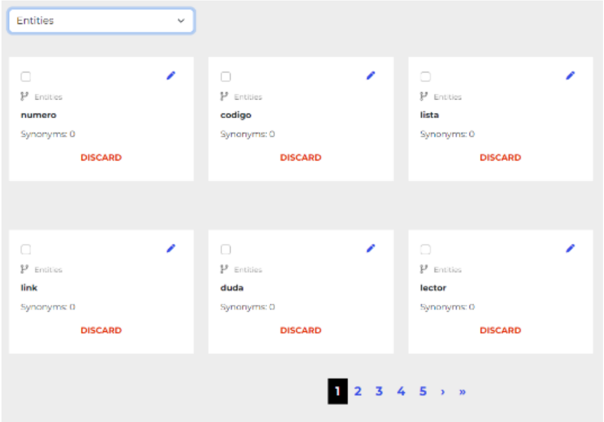
□ Intents
recibir
Examples: 3
DISCARD

□ Intents
no_reflejo
Examples: 1
DISCARD

□ Intents
no_tener
Examples: 7
DISCARD

1

**Fig. 12.** (2022). Table of Suggestions with the option to only show Intents.



Entities

□ Entities
numero
Synonyms: 0
DISCARD

□ Entities
codigo
Synonyms: 0
DISCARD

□ Entities
lista
Synonyms: 0
DISCARD

□ Entities
link
Synonyms: 0
DISCARD

□ Entities
duda
Synonyms: 0
DISCARD

□ Entities
lector
Synonyms: 0
DISCARD

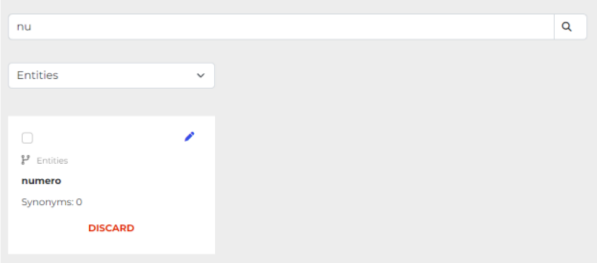
1 2 3 4 5 › »

**Fig. 13.** (2022). Table of Suggestions with the option to only show Entities.

You can make use of Axios and React Hooks to store lists of objects that are fetched from the MongoDB database in state variables. State variables are Hook functions that render and update in the lifecycle of the page. These Hook functions are preferred over normal variables, as the normals disappear instead of the Hooks that are preserved by React.

In this case, by calling an endpoint with the GET method in Flask on the back-end side, we get a list of objects that we can then identify and list on the page. By making use of and manipulating the lifecycle and state cycle of a Hook use State, variables can be updated and search filters can be performed on a list.

So, depending on whether the list is, whether it's a list of intents, a list of entities, or both, you can do a search function on whatever is listed on the page and then update that list that you render.

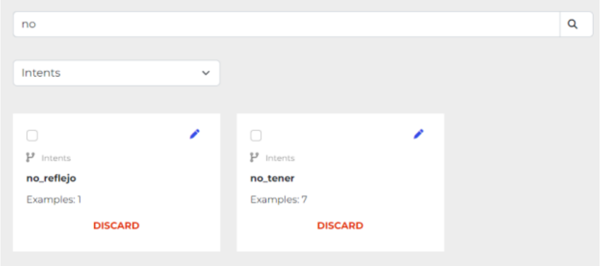


nu

Entities

□ Entities
numero
Synonyms: 0

**Fig. 14.** (2022). Searching for "a" among the list of entities.



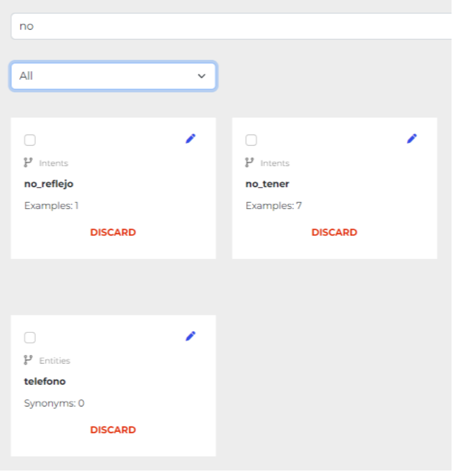
no

Intents

□ Intents
no_reflejo
Examples: 1

□ Intents
no_tener
Examples: 7

**Fig. 15.** (2022). Searching for "no" among the list of intentions.



no

All

□ Intents
no_reflejo
Examples: 1

□ Intents
no_tener
Examples: 7

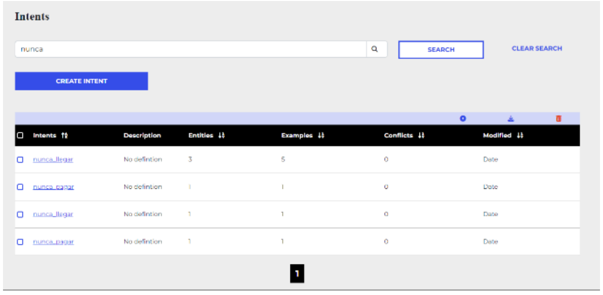
□ Entities
telefono
Synonyms: 0

**Fig. 16.** (2022). Searching for "no" among the list of intentions and entities.

Likewise, when you search or select an option to display, a pagination is updated that limits the number of objects that render on the page.

The dynamic search bar allows you to search for intents or entities by name, which helps to filter the intentions or entities that need to be viewed and then make a change to the database.

By using dynamic document search, you can access all the functions that an intent or entity has. Whether it's editing, viewing information, creating or deleting from the database.

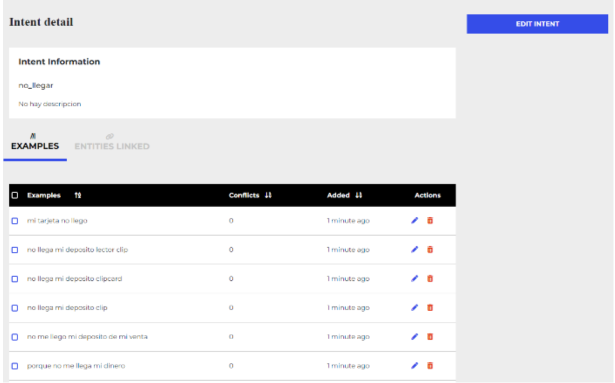


Intents

nunca

| ☐ | Intents | Description | Entities | Examples | Conflicts | Modified |
|---|---|---|---|---|---|---|
| ☐ | nunca_llegar | No defintion | 3 | 5 | 0 | Date |
| ☐ | nunca_pagar | No defintion | 1 | 1 | 0 | Date |
| ☐ | nunca_llegar | No defintion | 1 | 1 | 0 | Date |
| ☐ | nunca_pagar | No defintion | 1 | 1 | 0 | Date |

**Fig. 17.** (2022). Searching for "never" in the intentions section.



Intent detail

Intent Information

no_llegar

No hay descripcion

| ☐ | Examples | Conflicts | Added | Actions |
|---|---|---|---|---|
| ☐ | mi tarjeta no llego | 0 | 1 minute ago | |
| ☐ | no llega mi deposito lector clip | 0 | 1 minute ago | |
| ☐ | no llega mi deposito clipcard | 0 | 1 minute ago | |
| ☐ | no llega mi deposito clip | 0 | 1 minute ago | |
| ☐ | no me llego mi deposito de mi venta | 0 | 1 minute ago | |
| ☐ | porque no me llega mi dinero | 0 | 1 minute ago | |

**Fig. 18.** (2022). Visualizing the details of the intent named "no_llegar".

# 4 Conclusions

In conclusion, with everything that was mentioned in this article, the progress of the project has allowed us to generate solutions, front-end and methods of performing dynamic functions on the chatbot that help us fulfill the purpose of the project in developing a Natural Language Processing (NLP) platform.

Making use of various technologies such as ReactJS, Javascript, Python, MongoDB, Flask, and IBM Watson. In conjunction with React components and libraries such as Material UI (MUI), Hooks, and Axios, a platform is developed that is still in process for generating, creating, editing, visualizing, linking, and deleting intents, entities, and suggestions.

# References

1. NDS CONGNITIVE LABS: Rapid, cost-effective implementation of digital technologies. https://ndscognitivelabs.com/company-page/ (2022)
2. Burns, S.: Contextual entities with IBM Watson assistant: IBM Watson assistant. IBM watsonx Assistant. Accessed: https://medium.com/ibm-watson/contextual-entities-with-ibm-watson-assistant-f41b2e0ca82e (2022)

3. IBM: Assistant: Virtual agent. IBM. https://www.ibm.com/products/watson-assistant (2022)
4. MongoDB: The developer data platform. MongoDB. https://www.mongodb.com/ (2022)
5. React: Using the state Hook. https://reactjs.org/docs/hooks-state.html (2022)
6. Axios: Getting started. https://axios-http.com/docs/intro (2022)
7. Duraj, M.: How to build a chatbot using IBM Watson. https://www.pluralsight.com/guides/how-to-build-a-chatbot-using-ibm-watson (2022)

ISSN 1870-4069

# Automated Handwriting Analysis for Personality Traits Recognition Using Image Preprocessing Techniques

Diego A. Peralta-Rodríguez, Gerardo Lugo-Torres,
José E. Valdez-Rodríguez, Hiram Calvo

Instituto Politécnico Nacional,
Centro de Investigación en Computación,
Mexico

dperaltar1100@alumno.ipn.mx,
{glugot2022, jvaldezr2018, hcalvo}@cic.ipn.mx

**Abstract.** In this paper, we approach the problem of automated recognition and analysis of handwriting, applied in the context of personality trait estimation. Handwriting presents a singular and distinct expression for every individual, carrying valuable insights into the writer's personality and psychological traits. Despite several proposals already made to tackle this issue, there are still obstacles to overcome, such as the selection of algorithmic techniques for image quality enhancement. Our proposed methodology is based on analyzing handwritten text images, where we seek to identify patterns and features that allow us to infer specific personality traits. To achieve this, we have relied on the theoretical framework of the Big Five model, one of the most widely accepted models. The proposed methodology involved preprocessing images using a U-Net neural network and a convolutional layer-based architecture to classify personality traits.

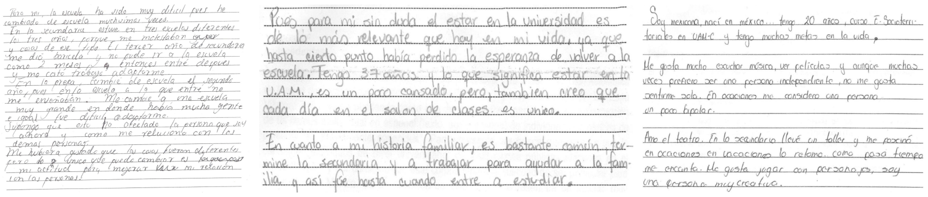
**Keywords:** Personality, traits, handwritten text, big five, image preprocessing.

## 1 Introduction

The automated recognition and analysis of handwritten or cursive writing is emerging as a highly fruitful field of research in neuropsychology, psychology, and computer science. This latter domain seeks to automate a traditionally performed by psychologists and specialists in calligraphic analysis, known as graphologists. While individual handwriting may share similarities or common traits with others, it remains distinctive and a personal hallmark.

As a result, analyzing the characteristics of strokes is a challenging task; on the contrary, it requires diverse techniques and tools. Exploring patterns and features in handwriting can provide valuable information about a person, including their age, gender, and motor control, and it may even serve as an early indicator of potential neurodegenerative disorders, given that writing involves coordinating hand movement, vision, and brain commands.

For instance, a brain injury, known as agraphia, could manifest as a loss of hand control. Furthermore, analyzing handwritten text allows us to identify personality traits, enabling us to understand, to some extent, a person's moods or motivations.



Para mi, la escuela ha sido muy dificil pues he cambiado de escuela muchisimas veces.
En la secundaria estuve en tres escuelas diferentes los tres años, porque me molestaban y cosas de ese tipo. El tercer año de secundaria me dió varicela y no pude ir a la escuela como 2 meses, y entonces entré despues y me costó trabajo adaptarme.
En la prepa, cambie de escuela el segundo año, pues en la escuela a la que entré no me enseñaban. Me cambie a una escuela muy grande en donde había mucha gente e igual fue dificil adaptarme.
Supongo que esto ha afectado la persona que soy ahora y como me relaciono con las demas personas.
Me hubiera gustado que las cosas fueron diferentes pero lo unico que puedo cambiar es mi actitud para mejorar mi relación con las personas.

Pues para mi sin duda el estar en la universidad es de lo más relevante que hoy en mi vida, ya que hasta cierto punto había perdido la esperanza de volver a la escuela. Tengo 37 años y lo que significa estar en la U.A.M., es un poco cansado, pero, tambien creo que cada día en el salon de clases es unico.

En cuanto a mi historia familiar, es bastante común, termine la secundaria y a trabajar para ayudar a la familia, y así fue hasta cuando entre a estudiar.

Soy mexicana, nací en méxico... tengo 20 años, curso E. Socioterritoriales en UAM-C y tengo muchas metas en la vida.

Me gusta mucho escuchar música, ver peliculas y aunque muchas veces prefiero ser una persona independiente, no me gusta sentirme sola. En ocaciones me considero una persona un poco bipolar.

Amo el teatro. En la secundaria llevé un taller y me fascinó en ocaciones en vacaciones lo retomo.. como pasa tiempo me encanta. Me gusta jugar con personajes, soy una persona muy creativa.

**Fig. 1.** Example of images included in the HWxPI dataset [12].

In psychology, numerous definitions have been proposed to explain the concept of personality. However, one of the most accepted is the proposition put forth by Allport, who defines personality as a dynamic organization within the individual of psychophysical systems that create characteristic patterns of behavior, thoughts, and feelings [1]. On the other hand, the evolution of personality concepts and components has led to the development of the so-called trait theory, wherein the trait is the central notion of personality psychology.

Traits are considered continuous entities along which individuals vary. We cannot directly observe a trait, but we can infer a person's level of trait based on their behavior. Several models and psychological tests within the trait theory attempt to describe the essential traits that define personality. One of the most robust models we rely on for this work is the Big Five personality model [2]. This model posits that personality can be summarized into five core factors:

1. Extroversion,
2. Agreeableness,
3. Conscientiousness,
4. Neuroticism,
5. Openness to Experience.

This paper addresses the problem of estimating personality traits from handwritten texts using Deep Learning techniques, leveraging the capabilities of Deep Learning, and specifically employing Convolutional Neural Networks (CNNs). The dataset used consisted of digitised versions of autobiographical essays handwritten by university students.

As for the methodology, this consisted of doing prior work on image processing and cleaning them of noise, using various methods, both classical mathematical morphology techniques and an encoder-decoder type network such as U-Net. Once the images were cleaned of noise, a neural network with five convolutional layers and three densely connected layers was used to perform the classification.

While the results obtained were not outstanding compared to other results in the state of the art in terms of accuracy, it is important to highlight that considerable performance could be achieved using only images without the need to work with text processing. We believe this is significant in terms of the approach to personality trait estimation using handwritten text. The rest of this paper is organized into the following sections: Section 2 addresses the state of the art, Section 3 presents the methodology used, and

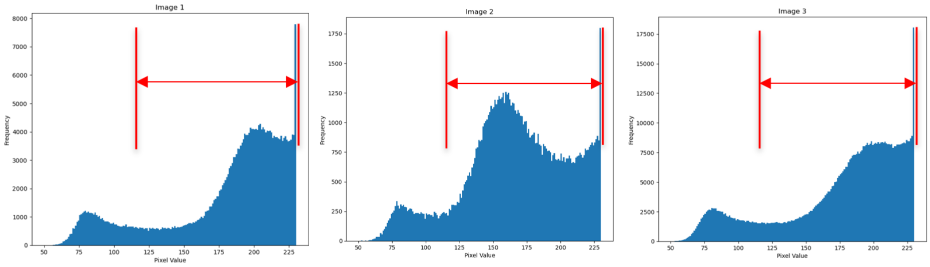


**Fig. 2.** Histograms of some selected images from the dataset were generated to analyze the distribution of pixel intensities in the handwritten texts. The defined range, which was carefully chosen, ensures that the text's details and readability are preserved during image processing and analysis.

some results derived from the proposed image preprocessing are shown, and Section 4 presents the obtained experimental results. Section 5 contains the conclusions. Finally, Section 6 briefly presents the limitations that arise when tackling the problem of trait estimation, and we discuss them.

## 2 State-of-the-Art

The following is a review of state-of-the-art papers related to identifying personality traits from handwritten texts, focusing exclusively on studies that utilized the same dataset as ours. Valdez-Rodríguez et al.[3] propose a Convolutional Neural Network (CNN) called DeepWriter, based on the AlexNet model. This architecture uses five convolutional layers with Rectified Linear Unit (ReLU) activation function to extract features from the images, such as the shape of the strokes in each author's letters.

Subsequently, two fully connected layers with hyperbolic tangent activation functions are included for classification, and finally, an output layer with two neurons is added. The model was classified using the Area Under the Curve (AUC) metric, which describes the classifier's ability to distinguish between different classes. Additionally, Costa et al. [4] present a method that combines features extracted from handwritten text images and their transcriptions.

For the analysis of transcriptions, the representation of a bag-of-words model is extracted considering a TF-IDF (Term frequency-Inverse document frequency) scheme. For feature extraction, they rely on the Linguistic Inquiry and Word Counter (LIWC) software, which is a resource that, given a text, analyzes the proportion of words that reflect different emotions and psycholinguistic aspects that can be inferred from texts [5]. As for the analysis of the essay images, the characters were segmented.

For this task, the datasets of EMNIST [6] and Chars74 [7] were trained with a Faster R-CNN model. Finally, they relied on Support Vector Machines (SVM) for classification. They used the AUC as an evaluation measure. Mekhaznia et al. [8] propose an approach to evaluate personality traits through various handwriting characteristics. Their method involves extracting features from handwriting samples and classifying them using an artificial neural network (ANN) algorithm.

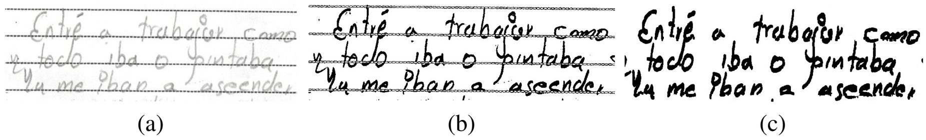


**Fig. 3.** a) Partial sample of image prior to filtering, b) Result after passing through the filter, and c) Result after applying morphological opening.

The primary experimental process consists of three phases: noise removal, construction of feature vectors, and feature classification. For noise removal, they adopted the idea by Dos Cardoso et al. [9], which introduces the concept of a stable path. A staff line is then considered an extended object of black pixels with a homogeneous width supported by a given shortest path. Removing staff lines involves replacing their pixels with white color and preserving other things that exceed the width.

Regarding constructing feature vectors, they retained the slants, writing direction, and ink trace features for experiments. The proposed classifier consists of three layers with a feed-forward architecture. The overall results for the Edge-Hinge Distribution and Run-Length Distributions features range from 50% to 60%.

# 3 Methodology

## 3.1 Dataset

The dataset used in this paper consists of 418 handwritten essays in Spanish written by undergraduate students from Mexico [10]. For each essay, two files are available: a manual transcription and a digitized image (Fig. 1). For this work, we solely utilized the images. Each essay has five classes corresponding to the Big Five personality traits.

The courses for each trait are represented as 1 and 0, corresponding to each trait's high and low poles, respectively. To assign each label in the dataset, the TIPI test was used [11]. This standardized and widely used instrument provides a set of norms to determine the direction of each trait among four classes: high, medium-high, medium-low, and low. For this dataset, the creators binarized the classes to 1 for subjects with high and medium-high traits and 0 for standard and medium-low traits.

## 3.2 Preprocessing

One of the main reasons for performing a dataset treatment was that, when reviewing the state-of-the-art works, it was observed that no attention was given to the ruled line removal process. Only the work of Mekhaznia et al. proposes a method to clean the images. Despite the emergence of convolutional neural networks, there must be more exploration in line removal and even in the general denoising of document images using neural network models.

In a recent study by Gold, a synthetic dataset was developed that creates lines and combines them in a sequence of handwritten word images using a proposed architecture with three convolutional layers. However, we encountered a problem with the dataset we obtained as it did not include corresponding lineless images.

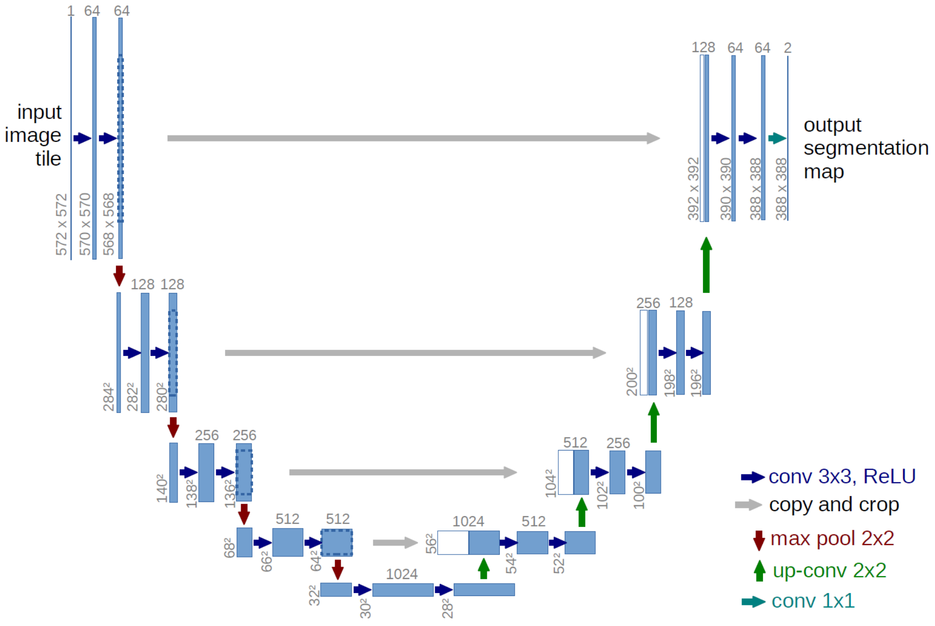


**Fig. 4.** U-Net architecture [13].

To solve this issue, we proposed an automated method for removing ruled lines, which is described below. First, we created images without lines by selecting a subset from the HWxPI dataset. We aimed to match the original images with their corresponding line-free versions, which would serve as labels for training the neural network model in line removal. We applied classical preprocessing techniques and mathematical morphology to construct this dataset. All images in the dataset were cropped to retain only the areas with text.

Next, we randomly selected 100 images, which were then converted to grayscale and processed with a Gaussian filter. This filter is typically used to reduce noise and flaws in images. By replacing each pixel with a weighted combination of its neighboring pixels, the filter smoothed the image and removed noise and minor variations in pixel intensity. We used a Gaussian filter with a kernel size of 3 x 3 and an auto-detector for the standard deviation.

Finally, we analyzed the histogram and pixel values that made up the lines in the essays to determine a suitable threshold. Defining a single threshold value for images with even minor variations can be complex and challenging. Factors such as the pressure applied while writing with a pencil or pen, continuous or dotted lines, and stains on the sheet can significantly alter pixel values.

Thus, removing lines entirely or partially from all images is impossible, as the image characteristics vary. Due to this variability, only a percentage of the images can be successfully processed for line removal. Once the necessary modifications were made, it was determined that the most suitable pixel range for thresholding was between 120 and 230 (Fig. 2).

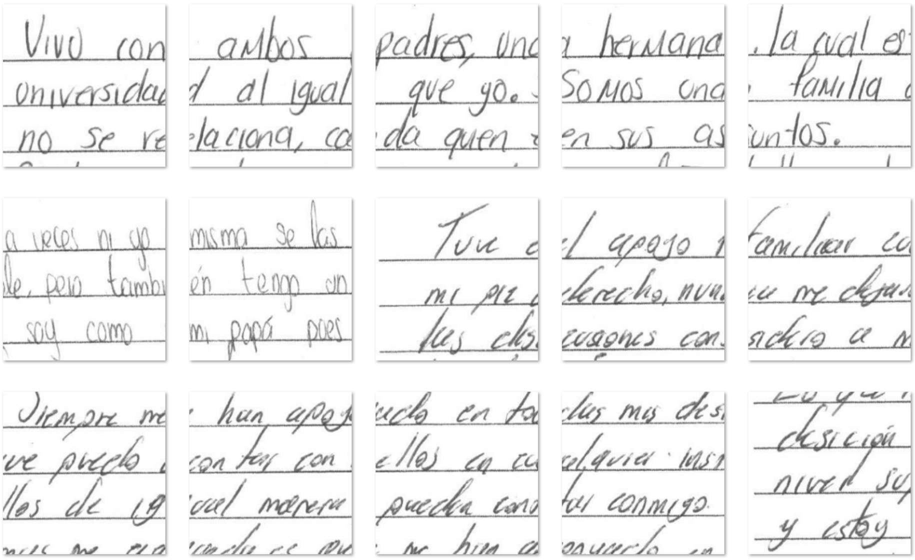


**Fig. 5.** Sample of Generated Patches.

Consequently, a filter was created to preserve only the grayscale values within that range. Out of the 100 images, applying the filter resulted in 40 images with almost complete line removal. Fig. 3 (a) and Fig. 3 (b) provide an example of a section of the image before and after passing through the filter. The filter effectively removed the filling in the lines, leaving only their outlines. To remove any remaining excess, an opening morphological operation was applied using Equation 1:

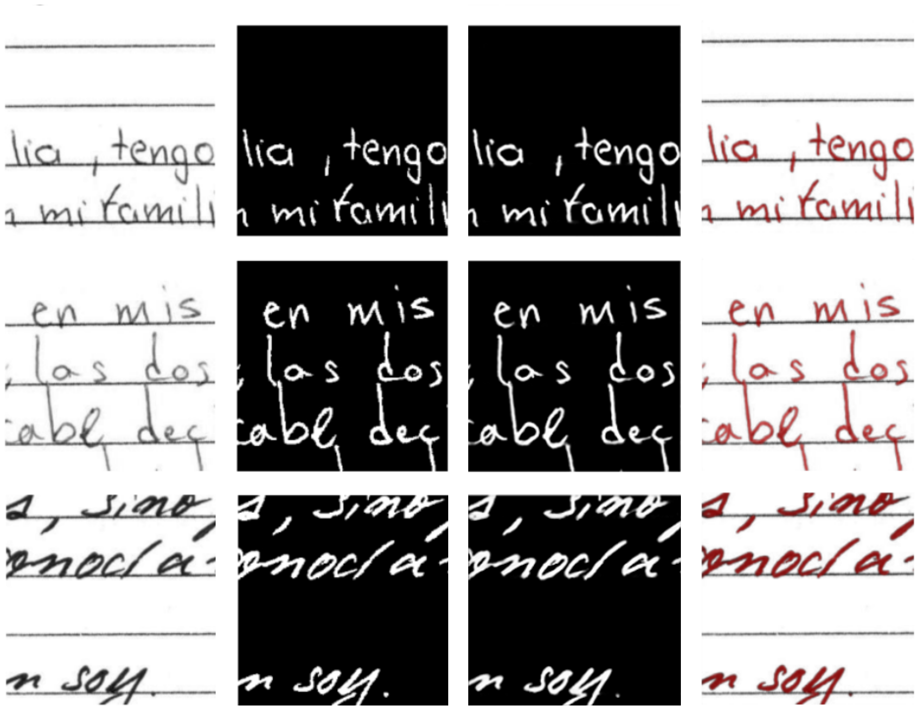
$$A \circ B = (A \ominus B) \oplus B. \tag{1}$$

This operation performs an erosion followed by dilation using the same structuring element for both procedures. Different structuring elements were experimented with, and the circular shape with a size of $3 \times 3$ worked best. The erosion completely removed the surplus contours, and the dilation helped preserve the letters' original structure (Fig. 3 (c)). Since some images had points of a particular area that could not be removed with erosion, an area opening was applied.

This functional filtering operation removes all connected components whose size in some pixels is smaller than the proposed threshold value. Ten additional images without ruled lines were added to the initial set of 40, but these were obtained by manually removing lines instead of using morphological operations. As a result, a small dataset composed of 100 images was generated: 50 original images and 50 line-free images.

### 3.3 U-Net

The U-Net architecture (Fig. 4) is advantageous because it can be trained with relatively few images, making it ideal for applications where a small dataset or real-time segmentation is needed [13].



(a) (b) (c) (d)

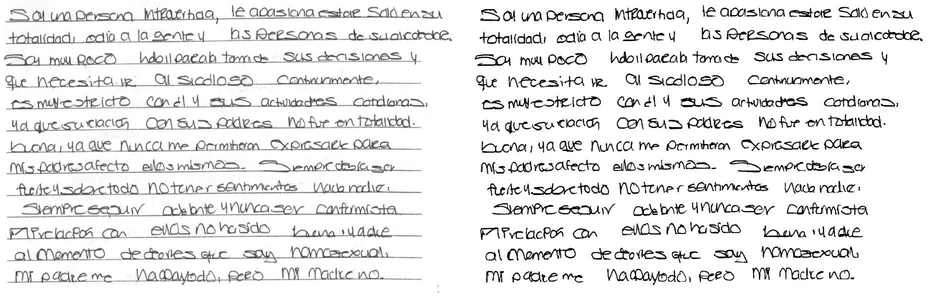
**Fig. 6.** Samples of some results: a) Original patches with lines, b) Ground Truth, c) Predictions, d) Intersection of the original patch with the prediction: the red color in the letters highlights a successful intersection between the original image and the prediction.

Due to this reason, we chose to work with this architecture, as the HWxPI image dataset obtained after line removal was limited. The U-Net is a deep Convolutional Neural Network (CNN) consisting of two main parts: the encoding pathway (encoder) and the decoding pathway (decoder). The encoding pathway is similar to the architecture of a typical CNN, where the input image is gradually reduced in size by applying convolutional layers and pooling.

On the other hand, the decoding pathway gradually increases the size of the image by applying convolutional layers and upsampling. Additionally, the U-Net uses a technique called ”skip connections,” which connects the encoding and decoding layers, allowing high-resolution information to be directly transmitted to the decoding layers, helping to avoid the loss of essential details in the image.

### 3.4 Training for Line Removal

Experiments were conducted with the images obtained from the HWxPI dataset. To expand the training set, dividing each image into patches of 256 × 256 pixels was proposed. Before this, each image was resized to multiples of 256 to avoid information loss and ensure capturing all regions of the image.



(a) Original Image (b) Image after processing.

**Fig. 7.** Result of U-Net model for an image.

As a result, a dataset consisting of 668 patches was generated (Fig. 5). Furthermore, data augmentation was performed, resulting in 64 additional images. This augmentation included rotation, mirroring, zooming, and stretching in height and width. One of the advantages of data augmentation is that it helps reduce overfitting by introducing variations in the training set, making the model less prone to memorizing the samples and enhancing its generalization capability. We used 64 filters, a dropout of 0.3, a batch size of 32, a definition of hyperparameters, and a sigmoid activation function. On the other hand, the metric used to evaluate the model was the Intersection over Union (IoU) (equation 2):

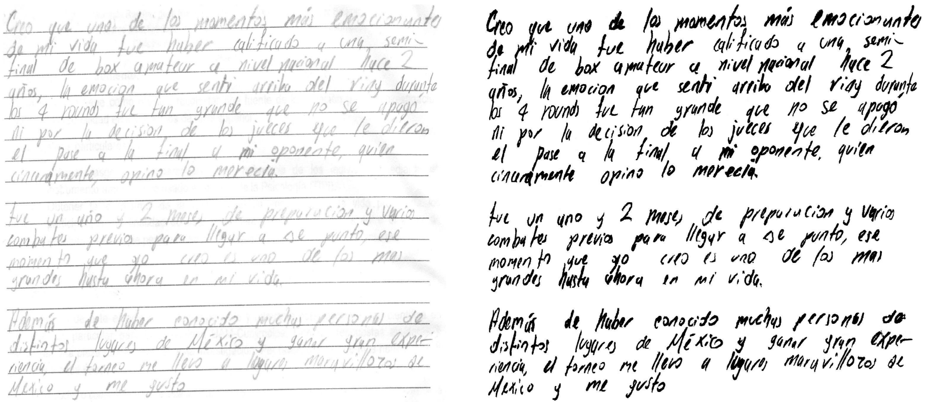
$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}}. \tag{2}$$

This metric involves dividing the intersection area between the model's predictions and the ground truth labels by the size of their union. It is widely used in segmentation tasks and allows measuring the similarity between the segmentation performed by the model and the accurate segmentation, where a value of 1 indicates a perfect segmentation and a value of 0 predicts an entirely erroneous segmentation. We employed the stochastic gradient descent algorithm and a binary cross-entropy loss function for model optimization.

During training, we obtained a loss value of 0.0667, while the Intersection over Union (IoU) coefficient resulted in a matter of 0.6847. When evaluating the model with the test set, we achieved an IoU value of 0.733. Once the model was established, all the images from the HWxPI dataset were utilized. A sample of the resulting patches are shown in Figure 6.

Below, we present some examples of the results in the preprocessing phase. The patches of each image were reassembled to restore each original image. Examples are shown in Figure 7 and 8. The use of an architecture like U-Net proved to be a valuable tool, allowing us to achieve good results in the removal of lines form handwritten texts without the need for additional exhaustive reconstruction work.

In only a few cases, a dilation followed by a morphological erosion was performed to close some gaps between letters. While this has demonstrated the potential of CNNs for line removal, it is essential to highlight that there are still areas that can be explored in future research. For instance, different network architectures can be proposed to further enhance result accuracy.



Creo que uno de los momentos más emocionante
de mi vida fue haber calificado a una semi-
final de box amateur a nivel nacional hace 2
años, la emocion que senti arriba del ring durante
los 4 rounds fue tan grande que no se apogo
ni por la decision de los jueces que le dieron
el pase a la final a mi oponente, quien
sinceramente opino lo merecia.

fue un año y 2 meses de preparacion y varios
combates previos para llegar a se punto, ese
momento que yo creo es uno de los mas
grandes hasta ahora en mi vida.

Además de haber conocido muchas personas de
distintos lugares de México y ganar gran expe-
riencia, el torneo me llevo a lugares maravillosos de
Mexico y me gusto

(a) Original Image

(b) Image after processing.

**Fig. 8.** Result of U-Net model for an image.

### 3.5 Proposed Architecture for Personality Traits Recognition

After completing the preprocessing step successfully, each image of the complete dataset was divided into patches of 256 x 256 pixels, generating a total of 11,263 patch images. Each of these patches was processed to remove the existing lines. Also, it was ensured that for each generated patch, a copy of the label originally assigned to the original image was replicated.

Then, we proceeded with the model design for personality traits estimation. In this dataset, each image can belong to five different classes. However, we decided to build a model to train courses by category, as these classes are not mutually exclusive. The model comprises five convolutional layers, followed by normalization and max pooling layers. Then, there are dropout layers to prevent overfitting. Finally, there are dense layers that produce the model's output (Fig. 9).

The architecture is shown below, where **d** represents dimensions; **k** denotes kernel size; **p** is the pool size, **s** represents stride, and **f** indicates filters. The first layer of dropout had a value of 0.3, and the remaining two layers had a value of 0.7.

The dataset consisting of 11,263 line-free patches was partitioned into 80% for training, 10% for validation, and 10% for testing purposes. The model was trained from scratch using the Adam optimizer with a learning rate of 0.0001, a binary cross-entropy loss function, and the binary accuracy metric, considering the binary nature of the classes. In addition, a filtering process was implemented to remove patches containing no text.

This refers to situations where the sum of all pixels in a patch was zero or contained minimal strokes that did not contribute relevant information, such as dots or letter traces. This process left out 430 patches that contained no information. To ensure consistency, we utilized a batch size of 16, and the training went on for 300 epochs. Notably, when increasing the number of epochs, we observed a tendency for the training process to overfit.

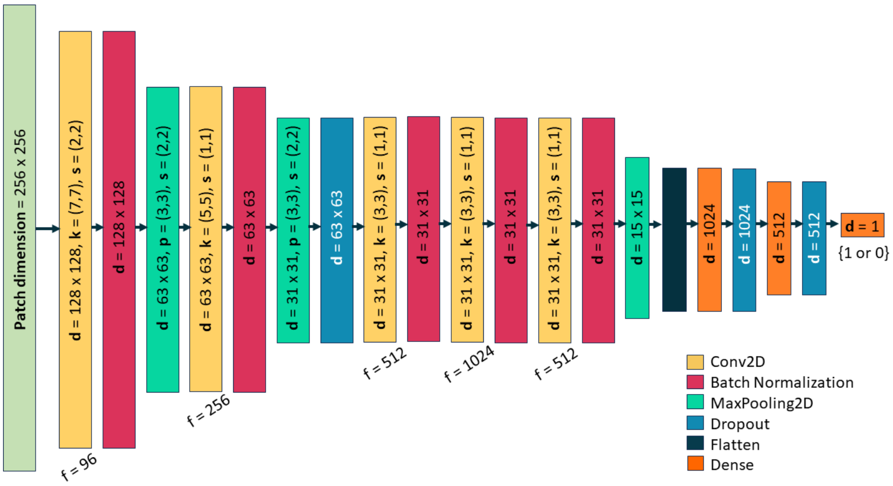

Patch dimension = 256 x 256
d = 128 x 128, k = (7,7), s = (2,2)
f = 96
d = 128 x 128
d = 63 x 63, p = (3,3), s = (2,2)
d = 63 x 63, k = (5,5), s = (1,1)
f = 256
d = 63 x 63
d = 31 x 31, p = (3,3), s = (2,2)
d = 63 x 63
d = 31 x 31, k = (3,3), s = (1,1)
f = 512
d = 31 x 31
d = 31 x 31, k = (3,3), s = (1,1)
f = 1024
d = 31 x 31
d = 31 x 31, k = (3,3), s = (1,1)
f = 512
d = 31 x 31
d = 15 x 15
d = 1024
d = 1024
d = 512
d = 512
d = 1
{1 or 0}
Conv2D
Batch Normalization
MaxPooling2D
Dropout
Flatten
Dense

**Fig. 9.** Proposed Architecture.

**Table 1.** Evaluation of our method.

| | **AUC** | **Precision (weighted avg)** | **Recall (weighted avg)** | **F1-score (weighted)** | **Accuracy** |
|---|---|---|---|---|---|
| Extraversion | **0.625** | **0.64** | **0.62** | **0.61** | **0.61** |
| Agreeableness | 0.59 | 0.58 | 0.59 | 0.59 | 0.58 |
| Conscientiousness | 0.60 | **0.63** | 0.59 | 0.57 | 0.58 |
| Neuroticism | 0.50 | 0.31 | 0.56 | 0.40 | 0.55 |
| Openness | 0.50 | 0.25 | 0.50 | 0.33 | 0.50 |
| **Average** | **0.56** | **0.48** | **0.57** | **0.50** | **0.56** |

## 4 Experimental Results

The classifier received and rated each patch separately. However, the purpose was to evaluate each complete image, not just its patches. To achieve this, the ratings of all the patches from each image were averaged to obtain the final rating for that image. The individual patch ratings were combined to get an overall rating for the entire image.

Various metrics were used to evaluate the results (Table 1) and the average of all rankings per metric was also obtained. The personality trait of Extraversion received the highest scores, surpassing 0.60 in all metrics. The personality trait of Conscientiousness received the second-highest scores. Conversely, neuroticism and openness received the lowest values.

In Table 2 and Table 3, we present the results from other state-of-the-art studies to facilitate a comparison with our results. In the work by Valdez et al. [3], although the final scores for each emotion are not shown, the average score across all classes was 0.5023. On the other hand, in Costa et al.'s work [4], the best results using a combination of Natural Language Processing techniques with image analysis (Shape) were 0.53 for extraversion and openness using LIWC.

**Table 2.** Results by Costa et. al [4].

| Features | Extraversion | Agreeableness | Conscientiousness | Neuroticism | Openness |
|---|---|---|---|---|---|
| LIWC | 0.51 | 0.49 | 0.47 | 0.45 | 0.54 |
| BoW | 0.52 | 0.50 | 0.49 | 0.53 | 0.50 |
| Shape | 0.50 | 0.50 | 0.57 | 0.45 | 0.53 |
| LIWC+BoW | 0.43 | 0.47 | 0.49 | 0.48 | 0.50 |
| LIWC+Shape | 0.53 | 0.47 | 0.48 | 0.44 | 0.53 |
| BoW+Shape | 0.50 | 0.50 | 0.56 | 0.48 | 0.52 |
| BoW+LIWC+Shape | 0.43 | 0.47 | 0.48 | 0.48 | 0.49 |

**Table 3.** Results by Valdez et. al [3].

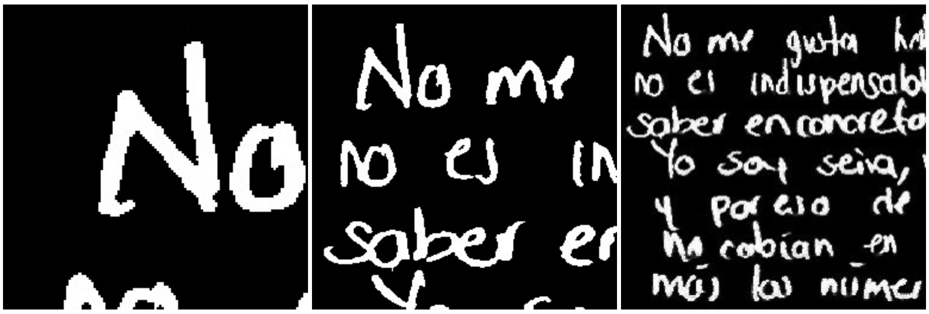
| | Baseline (validation set) | Validation set | Final evaluation |
|---|---|---|---|
| AUC | **0.5014** | **0.5023** | **0.5009** |
| F1 (multiclass) | 0.2629 | 0.1931 | 0.2025 |
| F1 (multilabel) | 0.2994 | 0.2916 | 0.2738 |
| BAC (multiclass) | 0.2091 | 0.1888 | 0.2011 |
| PAC (multiclass) | 0.0136 | 0.0136 | 0.0135 |
| PAC (multilabel) | 0.5028 | 0.5028 | 0.5063 |
| Regression ABS | -0.0004 | -0.0876 | -0.1384 |
| Regression R2 | -1.0008 | -1.1753 | -1.2769 |

For conscientiousness, they achieved 0.56 using BoW (Bag of Words). In contrast, using only images, their highest result was 0.57. It is worth noting that during training phase, when using patches of smaller and larger dimensions than 256 × 256, obtained by dividing each image, we obtained results equal or less than 0.40. In other words, our model was able to learn better from image samples that encompassed more words rather than focusing on isolated letters, which is interesting. To illustrate and have a better understanding of this, examples of the content covered by patches of larger and smaller dimensions are shown in Fig. 10.

A remarkable outcome of our work is that we achieved a maximum score of 0.62 in the AUC metric and a precision score of 0.61 solely through image analysis. This contrasts with other state-of-the-art works that obtained lower values using natural language processing, image analysis techniques, or even a combination of both. It is also important to note that there was no need for prior feature extraction, unlike other state-of-the-art works [15, 16] that performed an initial feature extraction process, such as handwriting inclination, letter size, writing pressure, and whether strokes were connected or not, among other aspects.

## 5 Conclusions

In this work, we have presented a method for estimating personality traits from handwritten text images. The method involved two stages. The first stage was preprocessing, which aimed to create line-free images from the HWxPI dataset.



(a) (b) (c)

**Fig. 10.** Example of the original image division using patches of different dimensions. a) 125×125 px, b) 256×256 px, and c) 512×512 px.

To achieve this, a subset of the original dataset was selected and the images were cleaned by cropping them to capture only the area where the text was present. Classic techniques were used, such as Gaussian filtering, grayscale conversion, thresholding, and mathematical morphology techniques like simple opening, area opening, and dilation. Once this set was cleaned, we had a paired dataset: images with lines alongside their respective line-free images. This small dataset was used to train the line removal model using a U-Net-type neural network.

All of the above had to be done almost from scratch, except for having the HWxPI dataset as a reference, as there is currently no dataset designed explicitly for the task of line removal in document images. The second stage was the selection of the architecture for trait identification. Different layer arrangements and hyperparameters were tested, but the best results were obtained using the architecture shown in Fig. 9. While our results could not surpass 0.64, we believe that they still managed to stand out compared to other results in the state of the art.

## 6 Limitations and Discussion

When comparing our results with those obtained by other studies that used the same dataset and were based on the same psychological evaluation model, the Big Five, we can see that surpassing the 0.5 barrier in accuracy or AUC is challenging. Why is this the case? It is a difficult question to answer, as it could depend on several factors, such as the dataset used. Even the dataset creators mention that personality identification is quite challenging with a small sample of handwritten texts. This is reflected in their proposed baseline, where scores range from 0.47 to 0.54 with the AUC metric [10].

However, Samsuryadi [14] provides an excellent compilation and analysis of works related to personality analysis from handwritten text. Although some studies are not included, and the compilation needs to be updated due to recent publications in this field over the past two years, a contrast can be observed among the results of studies that apply different methods and rely on different tools. For example, there is a significant leap in scores between studies that use different psychological models for personality evaluation and those that rely solely on graphology or even combine both tools.

Additionally, differences are observed in studies that use different datasets, apply various preprocessing techniques, utilize diverse architectures, and extract different features, among other factors. Undoubtedly, one of the future tasks in the area of personality estimation from handwriting is to establish a clear division between the methods to follow. For instance, it is essential to define which psychological evaluation models are the most robust to work with or to create a database with samples in different languages. This is necessary to conduct an objective assessment of the state-of-the-art, as without clear rules or guidelines, results that appear to be good may not be truly reliable.

## References

1. Allport, G. W.: Pattern and Growth in Personality. Holt, Reinhart and Winston (1961)
2. Laak, J.: Las cinco grandes dimensiones de la personalidad. Revista de Psicología, vol. 14, no. 2, pp. 129–181 (1969) doi: 10.18800/psico.199602.002
3. Valdez-Rodríguez, J. E., Calvo, H., Felipe-Riverón, E. M.: Handwritten texts for personality identification using convolutional neural networks. In: International Conference on Pattern Recognition, pp. 140–145 (2018) doi: 10.1007/978-3-030-05792-3_13
4. Costa, E. P., Villaseñor-Pienda, L., Morales, E., Escalante, H. J.: Recognition of apparent personality traits from text and handwritten images. In: International Conference on Pattern Recognition, pp. 146–152 (2018) doi: 10.1007/978-3-030-05792-3_14
5. Pennebaker, J. W., Francis, M. E., Booth, R. J.: Linguistic inquiry and word count (LIWC2001). LIWC2001 Manual (1999)
6. Cohen, G., Afshar, S., Tapson, J., Schaik, A.: EMNIST: Extending MNIST to handwritten letters. In: International Joint Conference on Neural Networks, pp. 2921–2926 (2017) doi: 10.1109/ijcnn.2017.7966217
7. de-Campos, T., Babu, B., Varma, M.: Character recognition in natural images. In: Proceedings of the Fourth International Conference on Computer Vision Theory and Applications, vol. 2, pp. 273–280 (2009)
8. Mekhaznia, T., Djeddi, C., Sarkar, S.: Personality traits identification through handwriting analysis. In: Mediterranean Conference on Pattern Recognition and Artificial Intelligence, pp. 155–169 (2021) doi: 10.1007/978-3-030-71804-6_12
9. Capela, A., Rebelo, A., Cardoso, J., Guedes, C.: Staff line detection and removal with stable paths. In: Proceedings of the International Conference on Signal Processing and Multimedia Applications, vol. 1, pp. 263–270 (2008) doi: 10.5220/0001937802630270
10. Ramírez, G., Villatoro, E., Ionescu, B., Escalante, H. J., Escalera, S., Larson, M., Müller, H., Guyon, I.: Overview of the multimedia information processing for personality and social networks analysis contest. In: International Conference on Pattern Recognition, pp. 127–139 (2018) doi: 10.1007/978-3-030-05792-3_12
11. Ahmed, A. O., Jenkins, B.: Critical synthesis package: Ten-item personality inventory (TIPI) - A quick scan of personality structure. MedEdPORTAL (2013) doi: 10.15766/mep_2374-8265.9427
12. Gold, C., Zesch, T.: Cnn-based ruled line removal in handwritten documents. In: International Conference on Frontiers in Handwriting Recognition, pp. 530–544 (2022) doi: 10.1007/978-3-031-21648-0_36
13. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical Image Computing and Computer-Assisted Intervention, pp. 234–241 (2015) doi: 10.1007/978-3-319-24574-4_28

14. Samsuryadi, S., Kurniawan, R., Mohamad, F. S.: Automated handwriting analysis based on pattern recognition: A survey. Indonesian Journal of Electrical Engineering and Computer Science, vol. 22, no. 1 (2021) doi: 10.11591/ijeecs.v22.i1.pp196-206
15. Gahmousse, A., Gattal, A., Djeddi, C., Siddiqi, I.: Handwriting based personality identification using textural features. In: International Conference on Data Analytics for Business and Industry: Way Towards a Sustainable Economy, pp. 1–6 (2020) doi: 10.1109/icdabi51230.2020.9325664
16. Nair, G. H., Rekha, V., Krishnan, M. S.: Handwriting analysis using deep learning approach for the detection of personality traits. Ubiquitous Intelligent Systems, pp. 531–539 (2021) doi: 10.1007/978-981-16-3675-2_40

ISSN 1870-4069

# Modeling the Depressive Mind: A Machine Learning Approach to Deciphering Beck's Cognitive Triad

Cesar Macias, Miguel Soto,
Marco A. Cardoso-Moreno, Hiram Calvo

Instituto Politécnico Nacional,
Centro de Investigación en Computación,
Mexico

{cmaciass2021, msotoh2021, mcardosom2021,
hcalvo}@cic.ipn.mx

**Abstract.** Mental and cognitive well-being is of paramount significance for human beings. Consequently, the early detection of issues that may culminate in conditions such as depression holds great importance in averting adverse outcomes for individuals. Depression, a prevalent mental health disorder, can severely impact an individual's quality of life. Timely identification and intervention are critical to prevent its progression. Our research delves into the application of Machine Learning (ML) techniques to potentially facilitate the early recognition of depressive tendencies. By leveraging the cognitive triad theory, which encapsulates negative self-perception, a pessimistic outlook on the world, and a bleak vision of the future, we aim to develop predictive models that can assist in identifying individuals at risk. In this regard, we selected The Cognitive Triad Dataset, which takes into account six different categories that encapsulate negative and positive postures about three different contexts: self context, future context and world context. Our proposal achieved great performance, by relying on a strict preprocessing analysis, which led to the models obtaining an accuracy value of 0.96 when classifying aspect contexts; whereas a value of 0.83 in accuracy was achieved under the aspect-sentiment paradigm.

**Keywords:** Cognitive triad inventory, depression detection, machine learning, classification, logistic regression, support vector machine, SVM.

## 1 Introduction

Depression is a potentially fatal mental disorder and is estimated to affect 3.8% of the world's population according to World Health Organization (WHO)[1]. It has devastating effects on brain health as the dynamic state of the cognitive, emotional and mental domains, affecting the integrity of the structure and functionality of the brain [2, 3]. For this reason, it is important to identify factors that can help prevent depressive symptoms and serve as markers of some type of permanent brain damage [7, 8]. Cognitive therapy based on Beck's Cognitive Triad has become an effective tool for treating depression by helping people identify, challenge and change their negative thought patterns.

[1] www.who.int/es/news-room/fact-sheets/detail/depression

pp. 107–120; rec. 2023-06-04; acc. 2023-09-01

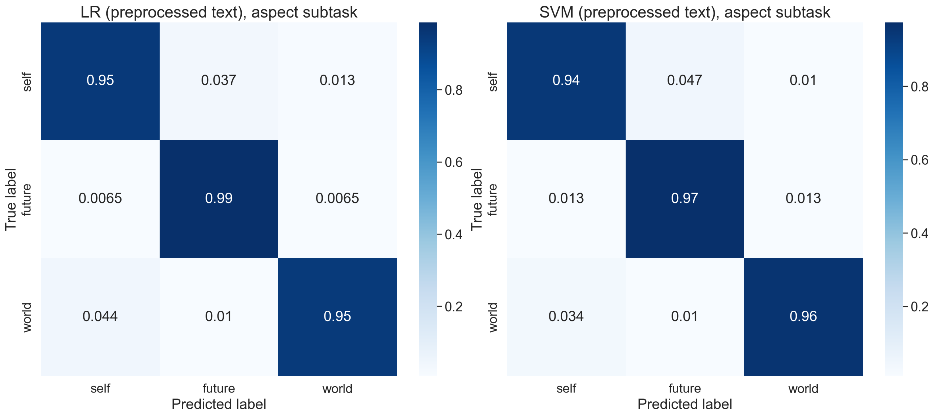
**Table 1.** Setup and results for set 1, subtask 1 (macro avg).

| Text feature | Model | Parameters | Accuracy | Precision | Recall | F1-score |
|---|---|---|---|---|---|---|
| Preprocessed | LR | `penalty="l2"` | **0.96** | **0.96** | **0.96** | **0.96** |
| | DT | `criterion="gini"` | 0.93 | 0.93 | 0.93 | 0.93 |
| | RF | `n_estimators=100` | 0.95 | 0.95 | 0.95 | 0.95 |
| | SVM | `kernel="rbf"` | **0.96** | **0.96** | **0.96** | **0.96** |
| No stopwords | LR | `penalty="l2"` | 0.69 | 0.69 | 0.69 | 0.69 |
| | DT | `criterion="gini"` | 0.64 | 0.64 | 0.64 | 0.64 |
| | RF | `n_estimators=100` | 0.68 | 0.68 | 0.68 | 0.68 |
| | SVM | `kernel="rbf"` | 0.70 | 0.70 | 0.70 | 0.70 |
| Lemmatized | LR | `penalty="l2"` | 0.95 | 0.95 | 0.95 | 0.95 |
| | DT | `criterion="gini"` | 0.93 | 0.93 | 0.93 | 0.93 |
| | RF | `n_estimators=100` | 0.95 | 0.95 | 0.95 | 0.95 |
| | SVM | `kernel="rbf"` | 0.95 | 0.95 | 0.95 | 0.95 |

By challenging and replacing these cognitive distortions, people can experience a reduction in depressive symptoms and improve their emotional well-being. Beck's Cognitive Triad is a central concept of the cognitive theory of depression developed by Aaron T. Beck [1]. This theory is based on the idea that people suffering from depression tend to have distorted and negative thought patterns that influence their emotional state. Beck's triad refers to three interrelated components of the following thought patterns:

- **Negative view of self (self-concept)**: People experiencing depression often have a negative perception of themselves. They feel that they are incompetent, unworthy or inadequate in various aspects of their lives. This negative self-perception can affect their self-esteem and self-confidence.
- **Negative view of the world (external world)**: Depressed people tend to view the world around them in a pessimistic and negative way. They may perceive their environment as hostile, threatening, or overwhelming. This negative view of the world can lead to feelings of hopelessness and lack of control over their environment.
- **Negative view of the future (future)**: Depressed people often have a pessimistic expectation about the future. They may anticipate difficulties and failures rather than positive possibilities. This negative outlook on the future contributes to feelings of hopelessness and apathy.

It stems from his approach to cognitive therapy of depression, which focuses on identifying and changing the negative and distorted thought patterns that contribute to the depressive experience. Beck proposed that these negative thought patterns are unrealistic and inaccurate, and that they influence a person's emotional state by maintaining and amplifying depression.



(a) LR (preprocessed text) (b) SVM (preprocessed text)

**Fig. 1.** Confusion matrices for set 1, subtask 1 best models.

It is of great importance because it provides a profound understanding of how thought patterns can influence a person's emotional state and contribute to depression. This theory helped change the way depression is approached and treated by focusing on the active role that thoughts and beliefs play in the depressive experience. Our approach is described in the following sections. This work is structured as follows.

In Section 2 a brief description of the related works in depression detection is given. In Section 3, the description of the datasets, the preprocessing, and feature extraction methods accompanied by the classification models implemented is given. Then, in Section 4 we describe the setup for the experiments and the results obtained after the submission of our predictions. Finally, in Section 5 we discuss our results and some ideas about how to improve them for future work.

## 2 State of the Art

Jere et al. [6], presented a dataset for the purpose of modeling Beck's Cognitive Triad (CTD) to model subjective symptoms of depression, such as negative view of self, future, and world. CTD is composed of $5,886$ messages, which are divided into six categories: self-positive (spos), world-positive (wpos), future-positive (fpos), self-negative (sneg), world-negative (wneg), and future-negative (fneg). This set was evaluated in two subtasks: aspect detection and aspect-specific sentiment classification. The main purpose of creating the dataset is to aid the understanding of Beck's Cognitive Triad Inventory (CTI) in people's social network postings.

Jere and Patil [5], use their previously created dataset and propose the following classical machine learning models: Decision Tree (DT), Random Forest (RF), Nayve Bayes (NB) and SVMs. They also implemented the following deep learning models: Graph Neural Networks (GNN), Long Short-Term Memory (LSTM), Bilateral Long Short-Term Memory (BiLSTM) and a Recurrent Neural Network (RNN). Their best model for aspect-based sentiment classification was the single-layer RNN, with a precision and F1-score of 0.857 and 0.858 respectively.

**Table 2.** Setup and results for set 1, subtask 2 (macro avg).

| Text features | Model | Parameters | Accuracy | Precision | Recall | F1-score |
|---|---|---|---|---|---|---|
| Preprocessed | LR | `penalty= "l2"` | 0.83 | 0.83 | 0.83 | 0.83 |
| | DT | `criterion="gini"` | 0.77 | 0.77 | 0.77 | 0.77 |
| | RF | `n_estimators=100` | 0.80 | 0.80 | 0.80 | 0.80 |
| | SVM | `kernel="rbf"` | **0.85** | **0.85** | **0.85** | **0.85** |
| No stopwords | LR | `penalty="l2"` | 0.76 | 0.76 | 0.76 | 0.76 |
| | DT | `criterion="gini"` | 0.68 | 0.68 | 0.68 | 0.68 |
| | RF | `n_estimators=100` | 0.73 | 0.73 | 0.73 | 0.73 |
| | SVM | `kernel="rbf"` | 0.76 | 0.76 | 0.76 | 0.76 |
| Lemmatized | LR | `penalty="l2"` | 0.82 | 0.82 | 0.82 | 0.82 |
| | DT | `criterion="gini"` | 0.76 | 0.76 | 0.76 | 0.76 |
| | RF | `n_estimators=100` | 0.80 | 0.80 | 0.80 | 0.80 |
| | SVM | `kernel="rbf"` | **0.85** | **0.85** | **0.85** | **0.85** |

For the sentiment recognition subtask, they obtained an accuracy and F1-score of 0.89 and 0.892 respectively with a multilayer RNN model. Finally, for aspect detection they obtained an accuracy and F1-score 0.957 and 0.956 respectively and again with a multilayer RNN. Gonzalez-Gomez et al. [4], analyzed data from two separate samples to assess associations between loneliness, social adjustment, depressive symptoms, and their neural correlates.

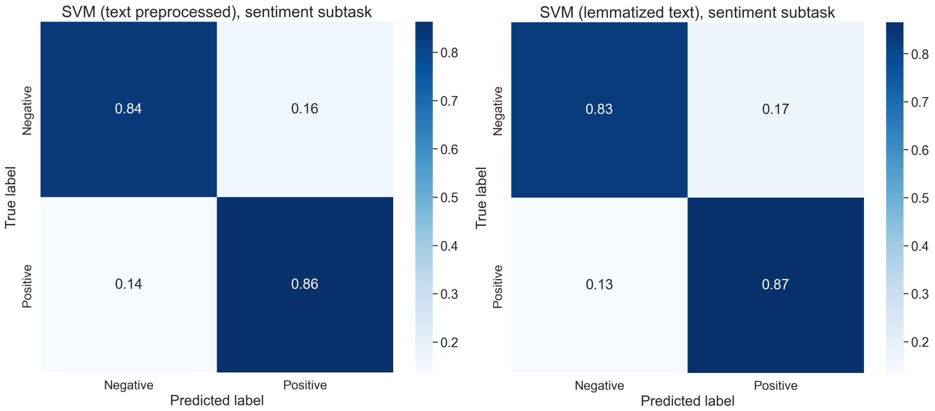
This was achieved by contacting study participants by telephone or social networks to complete various tapping scales on depressive symptomatology, loneliness and social adjustment and these results were assessed with the CTI and the University of California (UCLA) Loneliness Scale and found that after controlling for age, gender, and years of education only loneliness emerged as a significant predictor of depressive symptomatology.

## 3 Methodology

In this paper we will make use of the dataset collected by Jere et al. [6], in which we will perform an approach with machine learning models such as: Support Vector Machines, Logistic Regression, Decision Trees and Random Forest with different configurations. We will also make some adjustments to the documents of the original dataset by preprocessing them in order to obtain a better extraction of features for our models. Our final goal is to be able to compare our approach with the existing state-of-the-art results, we will perform the same subtasks that the state of the art has used to solve this problem, and we will also use the same evaluation metrics.

### 3.1 Models

**Support Vector Machine (SVM).** They are a supervised learning algorithm used for both classification and regression problems. An SVM searches for a hyperplane that best separates two different classes in the feature space.



(a) SVM (preprocessed text)

(b) SVM (lemmatized text)

**Fig. 2.** Confusion matrices for set 1, subtask 2 best models.

**Table 3.** Setup and results for set 2, subtask 1 (macro avg).

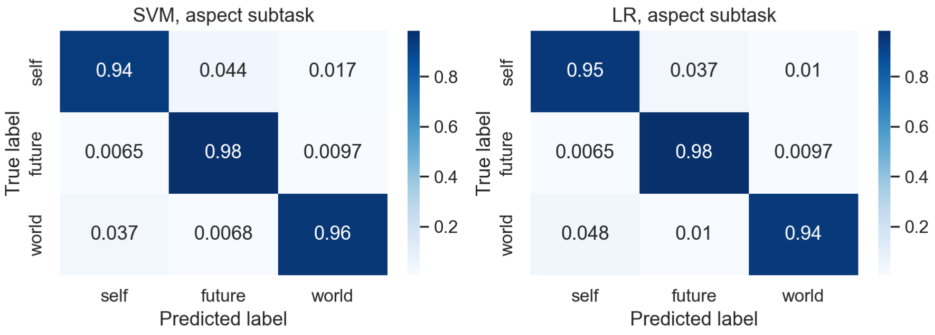
| Model | Parameters | Accuracy | Precision | Recall | F1-score |
|---|---|---|---|---|---|
| SVM | `kernel= "linear"` | 0.96 | 0.96 | 0.96 | 0.96 |
| LR | – | 0.96 | 0.96 | 0.96 | 0.96 |

The hyperplane is a geometric entity that is used to classify points into different categories. If the classes are not linearly separable in the original space, SVMs can use mathematical tricks to map the data to a higher dimensional space where they are linearly separable.

**Logistic Regression (LR).** It is a supervised learning algorithm used for classification and probability estimation in binary (two-class) or multi-class (more than two classes) problems. Logistic regression models the probability that an instance belongs to a particular class and uses the logistic function (also known as sigmoid function) to transform a linear combination of features into a value between 0 and 1, which is interpreted as the probability. Despite its name, it is mainly used for classification tasks rather than regression.

**Decision Trees (DT).** These are supervised learning algorithms used for classification and regression problems. They operate by recursively partitioning the feature space into smaller, more homogeneous subsets, with the objective of making decisions based on the hierarchical structure of the tree. At each node of the tree, a feature and a threshold value are selected to split the data into two branches (subsets) based on that feature. The selection of the splitting feature is made based on some criterion, such as information gain in the case of classification or variance reduction in the case of regression.

**Random Forest (RF).** They are a supervised learning algorithm used primarily for classification and regression tasks. They are a combination of multiple decision trees and are noted for their ability to handle both numerical and categorical data, as well as their ability to avoid overfitting and generate accurate predictions.



(a) SVM (b) LR

**Fig. 3.** Confusion matrices for set 2, subtask 1 best models.

**Table 4.** Setup and results for set 2, subtask 2 (macro avg).

| Model | Parameters | Accuracy | Precision | Recall | F1-score |
|---|---|---|---|---|---|
| SVM | `kernel= "linear"` | 0.85 | 0.86 | 0.85 | 0.85 |
| SVM | `kernel= "rbf"` | **0.86** | **0.86** | **0.86** | **0.86** |
| LR | – | 0.84 | 0.84 | 0.84 | 0.84 |

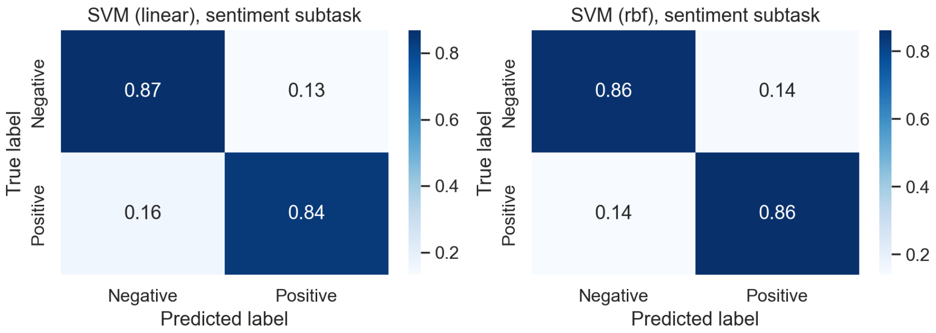
### 3.2 Text Preprocessing

Although the documents in the selected dataset are quite readable, there are some improvements that can be made to the text in order to improve the features that our machine learning model will be trained on. These enhancements include:

- **Line breaks**: All the line breaks were removed to obtain a single plain text.
- **Lowercase**: All the uppercase letters turned to lowercase.
- **Apostrophes**: All of them were removed.
- **Punctuation**: The characters used to punctuate the text (e.g., `. : , ;`) were removed.
- **Repeated characters**: When a word contains a character that repeats more than twice, the characters were trimmed into two repetitions (e.g., hellooooo → helloo).
- **Alphanumeric words**: The words composed by alphabetic and numerical characters like in leet speaking (e.g., h3ll0) were removed.

### 3.3 Feature Extraction

**Bag of words.** This model is a simplified representation of a text, where the bag represents the set of all words contained in a document collection. A single document is represented as a vector, where each dimension represents a word from the vocabulary obtained from the document collection and how many times the word appeared in a single document (TF).



(a) SVM (linear)

(b) SVM (rbf)

**Fig. 4.** Confusion matrices for set 2, subtask 2 best models.

**Table 5.** Results for set 2, aspect-sentiment CTD task (macro avg).

| Model | Accuracy | Precision | Recall | F1-score |
|---|---|---|---|---|
| Aspect-sentiment | 0.83 | 0.83 | 0.83 | 0.82 |

**TF·IDF.** This method makes use of the term frequency (TF), and the inverse document frequency (IDF). For the TF calculation, a bag of words is generated with the vocabulary of the set of all documents that are to be analyzed, then the total number of occurrences of the word is obtained. IDF is calculated as the logarithm of the quotient of the number of documents in which the analyzed word appears and the number of documents in which it appears in the analyzed set. Finally, the product between TF and IDF is performed for each of the words in the analyzed text, generating a vector of characteristics.

## 4 Experiments and Results

The solution proposed in this paper consists of two different approaches. The aspect-sentiment based classification and the multi-class scenario. These approaches are further described in their corresponding section.

### 4.1 Aspect-Sentiment Approach

For this approach, the main classification task was divided into two subtasks, the first one was the aspect identification subtask. For this task, we trained a model to detect whether the text was describing the author's thoughts about themselves (self), about future scenarios (future) or about the exterior (world). The second subtask was to detect the sentiment reflected by the author's in the text, whether they were positive or negative. For both sets of experiments, the dataset was redistributed to adequate it for the subtasks aim. The redistribution is described below.

**Subtask 1. Aspect identification.** A ternary dataset was generated, the class division was:

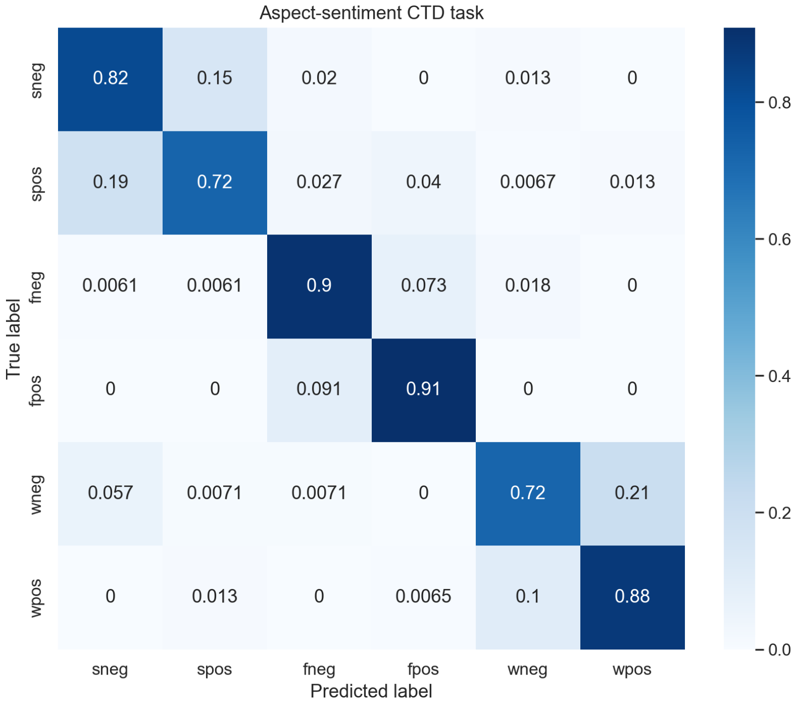


**Fig. 5.** Confusion matrix for set 2, aspect-sentiment CTD task.

- {self-negative, self-positive} → {self}
- {future-negative, future-positive} → {future}
- {world-negative, world-positive} → {world}

**Subtask 2. Sentiment classification.** For this subtask, the redistribution of the classes of the dataset was performed as follows:

- {self-negative, future-negative, world-negative} → {negative}
- {self-positive, future-positive, world-positive} → {positive}

**Experiments, set 1.** For this first phase of experimentation, we will perform experiments with the following machine learning models: LR, RF, DT and SVMs. The configuration to be used for these experiments will be the following: as tokenizer, we will use TF·IDF, with a minimum word occurrence frequency of 3 and a range of n-grams of 3. In order that these experiments can be replicated, it should be mentioned that the respective models of the `scikit-learn` library for Python with seed 42 were used.

**Subtask 1.** In the table 1 it can be seen that the best results with this configuration are LR and SVM with the specific feature that the text was preprocessed. We can also observe the behavior of the best models through their respective confusion matrices in the figure 1.

**Table 6.** Setup and results for set 1, multi-class (macro avg).

| Text features | Model | Parameters | Accuracy | Precision | Recall | F1-score |
|---|---|---|---|---|---|---|
| Preprocessed | LR | `penalty="l2"` | **0.79** | **0.80** | **0.79** | **0.79** |
| | DT | `criterion="gini"` | 0.69 | 0.69 | 0.69 | 0.69 |
| | RF | `n_estimators=100` | 0.76 | 0.76 | 0.76 | 0.75 |
| | SVM | `kernel="rbf"` | 0.79 | 0.79 | 0.78 | 0.78 |
| No stopwords | LR | `penalty="l2"` | 0.56 | 0.56 | 0.56 | 0.56 |
| | DT | `criterion="gini"` | 0.47 | 0.47 | 0.47 | 0.47 |
| | RF | `n_estimators=100` | 0.54 | 0.54 | 0.54 | 0.53 |
| | SVM | `kernel="rbf"` | 0.58 | 0.58 | 0.58 | 0.58 |
| Lemmatized | LR | `penalty="l2"` | **0.79** | **0.79** | **0.79** | **0.79** |
| | DT | `criterion="gini"` | 0.69 | 0.69 | 0.69 | 0.69 |
| | RF | `n_estimators=100` | 0.75 | 0.75 | 0.75 | 0.75 |
| | SVM | `kernel="rbf"` | **0.79** | **0.79** | **0.79** | **0.79** |

**Table 7.** Setup and results for set 2, multi-class ensemble model (macro avg).

| Model | Class | Kernel | Accuracy | Precision | Recall | F1-score |
|---|---|---|---|---|---|---|
| svm_sneg | self-negative | linear | 0.89 | 0.89 | 0.89 | 0.89 |
| svm_spos | self-positive | rbf | 0.85 | 0.85 | 0.85 | 0.85 |
| svm_fneg | future-negative | linear | 0.90 | 0.90 | 0.90 | 0.90 |
| svm_fpos | future-positive | rbf | 0.90 | 0.90 | 0.90 | 0.90 |
| svm_wneg | world-negative | rbf | 0.85 | 0.85 | 0.85 | 0.85 |
| svm_wpos | world-positive | linear | 0.93 | 0.93 | 0.93 | 0.93 |

**Subtask 2.** The results for this subtask can be seen in the table 2. For this second subtask, there are two models competing to be the best, however, both models are SVMs, with the difference being the treatment given to the text, one model was with preprocessed text and the other with lemmatized text. The confusion matrices for these models are shown in figure 2.

**Experiments, set 2.** After the first stage of experiments in the aspect-sentiment approach, we performed a second trial to improve the performance of those obtained in the first set.

**Subtask 1.** Once the classes were redistributed, the preprocessed texts were tokenized. The tokenizer and its parameters are the following: as tokenizer, we will use TF·IDF, with a minimum word occurrence frequency of 2, maximum document frequency of 0.9 and a range of n-grams of 2. Then, several models were trained and evaluated to test their performance.

After the evaluation, two models draw, obtaining the best performance. To make the experiments replicable, the seed for the random generations were set to 77. The best models setup and results are described in Table 3, their confusion matrices are shown in Figure 3.

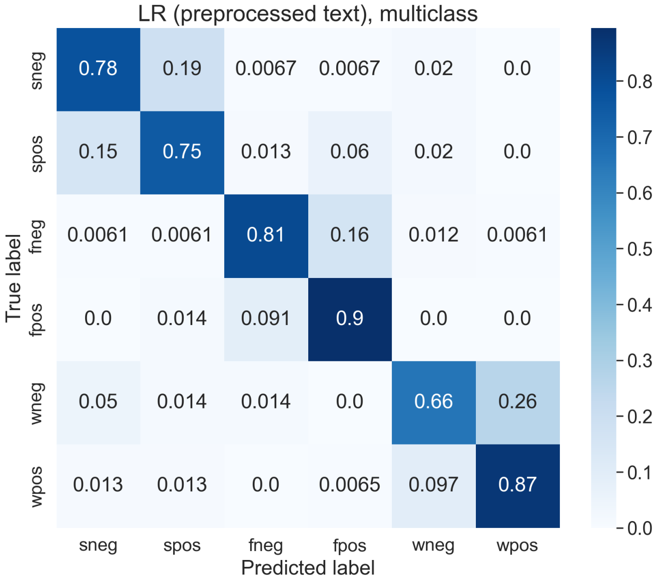


**Fig. 6.** Confusion matrix for set 1, LR (preprocessed text) multi-class.

**Table 8.** Results for set 2, multi-class SVM ensemble (macro avg).

| Model | Accuracy | Precision | Recall | F1-score |
|---|---|---|---|---|
| SVM ensemble | 0.77 | 0.77 | 0.77 | 0.77 |

**Subtask 2.** As in the aspect identification subtask, preprocessed texts were tokenized. The tokenizer ant its parameters are the following: as tokenizer, we will use TF·IDF, with a minimum word occurrence frequency of 2, maximum document frequency of 0.17 and a range of $n$-grams of 2.

In total, three models were trained and evaluated for this subtask, the model configuration and their results are described in Table 4, their confusion matrices are shown in Figure 4. Figure 4a corresponds to the SVM with linear kernel, Figure 4b shows the results for the SVM with `rbf` kernel.

**Aspect-sentiment classification.** Once both subtasks were completed, the cognitive triad of depression (CTD) classification task was performed. To accomplish it, the best models from the subtasks 1 and 2 were selected and put together:

– Subtask 1: SVM with linear kernel.

– Subtask 2: SVM with rbf kernel.

The results for this classification stage are shown in Table 5, and the confusion matrix is shown in Figure 5. This set of experiments revealed interesting things about the features and the models. For the aspect identification subtask, words with high frequency (those that are present in 90% of the documents or less) and low frequency ones (those repeated in at least two documents) are relevant for the model performance,

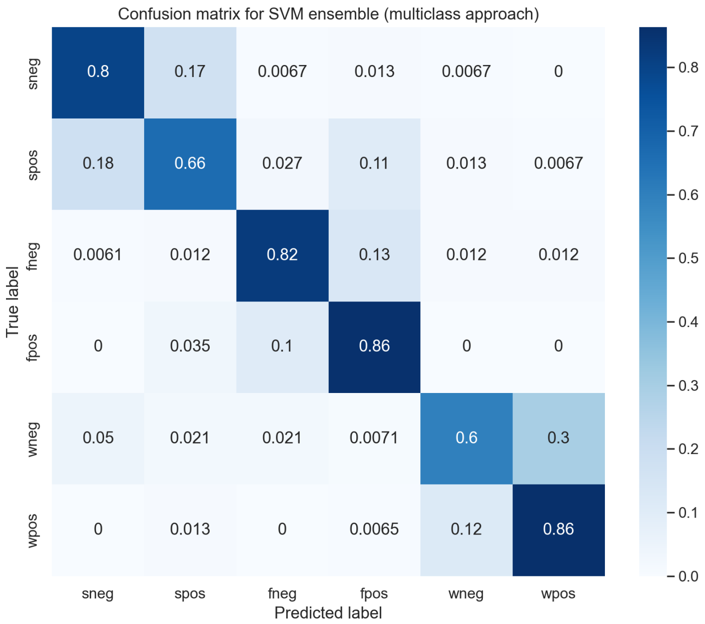


**Fig. 7.** Confusion matrix for set 2, multi-class SVM ensemble.

**Table 9.** Comparison, aspect identification task.

| Author | Model | Accuracy | Precision | Recall | F1-score |
|---|---|---|---|---|---|
| Jere et al. (2021) [6] | SVM | 0.77 | 0.78 | 0.77 | 0.78 |
| | RNN-C | **0.96** | **0.97** | 0.95 | **0.96** |
| Jere and Patil (2022) [5] | SVM | 0.77 | 0.77 | 0.76 | 0.77 |
| | LSTM | 0.90 | 0.90 | 0.90 | 0.90 |
| | RNN-AC | **0.96** | 0.96 | **0.96** | **0.96** |
| Our proposal | SVM | 0.96 | 0.96 | 0.96 | 0.96 |

but for the sentiment classification subtask, the most frequent words are irrelevant (those words repeated in more than the 17% of the documents are ignored) and even with that feature reduction, the models were able to obtain promising results. For both subtasks, the best models were SVMs that in the CTD classification task performed very good.

## 4.2 Multi-Class Approach

**Experiments, set 1.** The first experimentation phase for this subtask was based on the first experimentation phase for subtask 1. Thus, the models used as well as their corresponding configurations were exactly the same, the only difference will be that this approach will be a multi-class approach. The results of this phase are shown in the table 6, where we obtained three models with similar results: RL with preprocessed text, LR and SVM with lemmatized text.

**Table 10.** Comparison, sentiment classification task.

| Author | Model | Accuracy | Precision | Recall | F1-score |
|---|---|---|---|---|---|
| Jere et al. (2021) [6] | SVM | 0.78 | 0.79 | 0.78 | 0.79 |
| | RNN-C | **0.89** | **0.90** | 0.88 | **0.89** |
| Jere and Patil (2022) [5] | SVM | 0.77 | 0.78 | 0.77 | 0.77 |
| | LSTM | 0.87 | 0.87 | 0.87 | 0.87 |
| | RNN-AC | **0.89** | 0.89 | **0.89** | **0.89** |
| Our proposal | SVM | 0.86 | 0.86 | 0.86 | 0.86 |

**Table 11.** Comparison, aspect-sentiment CTD classification task.

| Author | Model | Accuracy | Precision | Recall | F1-score |
|---|---|---|---|---|---|
| Jere et al. (2021) [6] | SVM | 0.61 | 0.62 | 0.60 | 0.61 |
| | RNN-C | **0.86** | **0.86** | **0.86** | **0.86** |
| Jere and Patil (2022) [5] | SVM | 0.61 | 0.62 | 0.60 | 0.61 |
| | LSTM | 0.82 | 0.82 | 0.82 | 0.82 |
| | RNN-AC | **0.86** | **0.86** | **0.86** | **0.86** |
| Our proposals | Aspect-sentiment | 0.83 | 0.83 | 0.83 | 0.82 |
| | LR (preprocessed text) | 0.79 | 0.80 | 0.79 | 0.79 |
| | SVM-ensemble | 0.77 | 0.77 | 0.77 | 0.77 |

RL with the preprocessed text was chosen as the best model of all since LR with the lemmatized text at the end of the experimentation did not end up converging with the predetermined number of iterations, on the other hand the SVM model with the lemmatized text has a lower precision value. The confusion matrix for the best model is shown in Figure 6.

**Experiments, set 2.** After the first stage of experiments in the multi-class scenario, we performed a second experiment that consisted of a model ensemble. The ensemble consists of six SVMs trained to identify each of the six classes. To put the ensemble together, we first trained each of the models in a binary classification scenario.

To achieve this, the main dataset was divided into subsets that contained all the samples of the class to train, then we balanced the subsets with random samples of data of the other five classes in the same proportion (approximately 20% of the size of the class to balance), obtaining six subsets to train and test the models of the ensemble.

For each of the models a different tokenizer was fitted to tokenize the preprocessed text, the parameters used to fit the tokenizers were the same, a TF·IDF vectorizer from `scikit-learn` with an N-gram range of unigrams and bigrams, trough experimentation, we realized that unique words (those that appear just in one document) does not affect the performance of the models and that those words with high frequencies neither.

The final setup is the following: as tokenizer, we will use TF·IDF, with a minimum word occurrence frequency of 2, maximum document frequency of 0.25 and a range of n-grams of 2. The setup and results for each of the ensemble models is described in Table 7. To make the experiments replicable, the random generation seeds were set to 77.

Finally, when the models were put all together to build the ensemble, the results shown in Table 8 were obtained. The confusion matrix for the SVM ensemble model is displayed in Figure 7. The results obtained from this approach revealed that the multi-class approach is promising, further experimentation is required to outperform the results obtained from the aspect-sentiment approach.

### 4.3 Comparison with the State of the Art

Here, a comparison between the results obtained from our proposal and those from the state of the art is provided. The results obtained from the aspect identification tasks are shown in Table 9, those obtained from the sentiment classification task in Table 10, and the CTD classification are shown in Table 11. The comparison was made between ML and DL models on purpose, from the tables above, it can be inferred that although our ML proposals did not outperform the DL models are very competitive, overpassing the ML models proposed in the literature, and even reaching the DL models performance in some cases like the aspect identification task. The multi-class approach got some very good results, providing a new baseline for future proposals in this approach.

## 5 Discussion and Future Work

The results obtained in this paper suggest that ML algorithms are suitable for classifying text sequences based on two different characteristics that are simultaneously present under the Beck's Cognitive Triad, i.e., aspect analysis, whether it is self, world or future focused; and sentiment analysis, to determine if what was said is positive or negative with respect to the corresponding aspect. In this regard, our aspect-sentiment approach, which consists of two models, each one of them responsible of classifying either aspect or sentiments, to subsequently conjugate their results to deliver a final label, showed to be the most appropriate proposal for the task in question.

Additionally, we consider that these results are a consequence of the preprocessing pipeline applied to the data, which was based on our observations on the tasks after an in depth analysis. Therefore, we believe that data processing, even though lately, has received little to no attention, is still a procedure of vital importance, since it allows algorithms to learn on specific properties of the input data.

Furthermore, a new approach to this problem, the multi-class approach, was proposed. This approach trains directly on the corpus, without the need to divide the main task into subtasks. Despite this strategy, the proposed models showed good performance, outperforming those ML models proposed in the literature, setting the baseline for future attempts in this approach.

Although in this paper we focused exclusively on traditional ML algorithms, due to the evidence in the literature, we consider that further research needs to be carried on taking into account Deep Learning (DL) architectures, such as Recurrent and Convolutional Neural Networks; furthermore, the preprocessing stages need to keep playing an important role, since not only linguistic features need to be considered, but under the DL paradigm, the type of data a given architecture is prone to is also of great relevance for the model's performance.

## References

1. Beck, A. T., Powles, W. E., Alford, B. A.: Depression: Causes and treatment. American Journal of Clinical Hypnosis, vol. 16, no. 4, pp. 281–282 (1972) doi: 10.1080/00029157.1974.10403697
2. Chapman, S. B., Fratantoni, J. M., Robertson, I. H., D'Esposito, M., Ling, G. S. F., Zientz, J., Vernon, S., Venza, E., Cook, L. G., Tate, A., Spence, J. S.: A novel BrainHealth index prototype improved by telehealth-delivered training during COVID-19. Frontiers in Public Health, vol. 9 (2021) doi: 10.3389/fpubh.2021.641754
3. Chen, Y., Demnitz, N., Yamamoto, S., Yaffe, K., Lawlor, B., Leroi, I.: Defining brain health: A concept analysis. International Journal of Geriatric Psychiatry, vol. 37, no. 1 (2022) doi: 10.1002/gps.5564
4. Franco-O´Byrne, D., Gonzalez-Gomez, R., Morales Sepúlveda, J. P., Vergara, M., Ibañez, A., Huepe, D.: The impact of loneliness and social adaptation on depressive symptoms: Behavioral and brain measures evidence from a brain health perspective. Frontiers in Psychology, vol. 14 (2023) doi: 10.3389/fpsyg.2023.1096178
5. Jere, S., Patil, A. P.: Aspect-based sentiment classification for detecting the cognitive triad mechanism of depression. Journal of Computer Science, vol. 18, no. 12, pp. 1144–1158 (2022) doi: 10.3844/jcssp.2022.1144.1158
6. Jere, S., Patil, A. P., Shidaganti, G. I., Aladakatti, S. S., Jayannavar, L.: Dataset for modeling Beck's cognitive triad to understand depression. Data in Brief, vol. 38 (2021) doi: 10.1016/j.dib.2021.107431
7. Jiang, L., Zhang, S., Wang, Y., So, K.-F., Ren, C., Tao, Q.: Efficacy of light therapy for a college student sample with non-seasonal subthreshold depression: An RCT study. Journal of affective disorders, vol. 277, pp. 443–449 (2020) doi: 10.1016/j.jad.2020.08.055
8. Zhang, B., Liu, S., Liu, X., Chen, S., Ke, Y., Qi, S., Wei, X., Ming, D.: Discriminating subclinical depression from major depression using multi-scale brain functional features: A radiomics analysis. Journal of Affective Disorders, vol. 297, pp. 542–552 (2022) doi: 10.1016/j.jad.2021.10.122

ISSN 1870-4069

# Understanding the Limits of Average Aggregation in Federated Learning Scenarios

Sergio Pérez-Picazo, Hiram Galeana-Zapién[1],
Edwyn Aldana-Bobadilla

Instituto Politécnico Nacional,
Centro de Investigación y de Estudios Avanzados,
México

{sergio.perez.picazo, hiram.galeana,
edwyn.aldana}@cinvestav.mx

**Abstract.** The prevalent design process nowadays for producing machine learning models requires large datasets that contain different types of data (numerical, categorical, ordinal) depending on the use case or application interest. However, increased Internet users' concerns regarding data privacy might significantly impact the data collection process in specific machine applications, particularly those related to sensitive data like healthcare services. In this regard, the feasibility of federated learning has been investigated, commonly using the seminal Federated Averaging (FedAvg) algorithm to produce a generalized model from the individual models shared by the users through an aggregation process. In this context, this paper aims to contribute to understanding, to a certain extent, the scope of model averaging (used by FedAvg) to identify the cases in which its application is feasible. For this, we adopt a generic federated scenario in which we abstract the selection and configuration stages to focus on the aggregation process. Due to the diversity of possible architectures to perform the aggregation process, we consider three generic architecture families. Through simulation, we quantify the precision achieved by each method under different conditions, such as the number of participants, the type of data used in local learning, etc.

**Keywords:** Federated learning, model aggregation, neural networks.

## 1 Introduction

In the last decade, Internet cloud services have allowed users to store and process large amounts of data that can be accessed anytime, anywhere, from personal computers to mobile devices such as smartphones, wearables, home assistants, etc. By gathering data in the cloud, new business models based on information extraction using machine learning (ML) algorithms can be created, transforming decision-making into knowledge-based decision processes. The design process of machine learning algorithms involves training a learning model using a dataset stored in cloud computing facilities. While this is the commonly followed approach nowadays, a growing concern exists regarding data privacy, where users might struggle to share their data for machine learning purposes with a centralized entity on the Internet. This could be the case in specific scenarios dealing with sensitive data, such as healthcare.

pp. 121–136; rec. 2023-06-04; acc. 2023-09-01

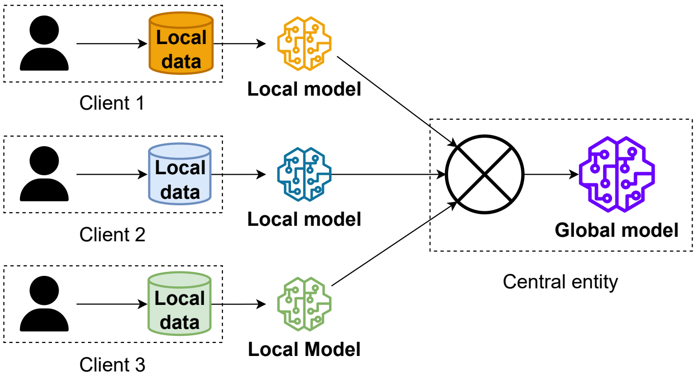


**Fig. 1.** Federated learning system.

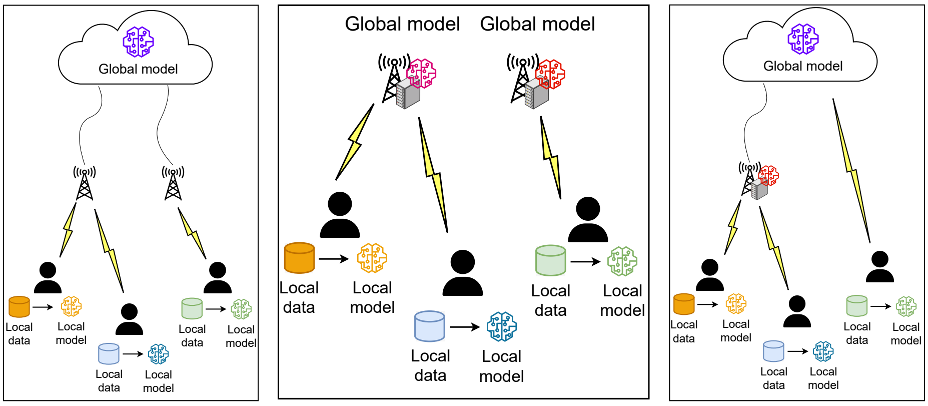
Based on this, Google introduced the term federated learning (FL) in 2017, a brand-new paradigm for machine learning that tackles a few concerns in terms of data privacy, network bandwidth restrictions, and device capabilities (computing, energy, and communication resources). FL emerged as a solution where various users (devices, organizations, etc.) can train a common global model jointly. The training process in federated learning incorporates two stages:

- Each client (device or organization) uses the collected data to train a local model, which is sent to a central entity.
- A centralized entity receives all the local models from clients and executes an aggregation process to create a global model.

At the same time, Google unveiled the first federated learning algorithm called Federated Averaging (FedAvg) [2] based on a progressive training process where model averaging is applied through a three-stage structure deployed in a cloud architecture. For each learning round: 1) a set of clients is selected to participate; 2) each selected client computes an update to the current global model by the server, and only the update is transmitted; and 3) the cloud server aggregates all received local models.

Meanwhile, for the common global model, FedAvg established an aggregation method based on average. The parameters of local models are averaged element-wise with weights proportional to the sizes of the client datasets. The model averaging method emerged as the standard aggregation method used most frequently in academia because of FedAvg's performance on numerous classification and prediction problem tasks. In addition, the development of FL algorithms, mostly FedAvg variants, indicated room for improvement in several areas.

For instance, certain works, such as [5, 4], have investigated how to optimize wireless networks for FL implementation and address issues with wireless communication such as orientation, mobility, and intrusion detection [7]. Other FL algorithms are focused on enhancing the FedAvg structure. For instance, improving the selection stage through better decision-making about which clients could be considered participants in the aggregation process.



(a) Cloud-based FL. (b) Edge-based FL. (c) Hybrid FL.

**Fig. 2.** Deployment strategies for a FL system.

Meanwhile, the configuration stage is refined in the model training process, specifically by optimizing the clients' computational resources. However, aggregation is barely explored because many FL algorithms have opted for the classic averaging method for model aggregation. In this sense, we identify a need to explore other aggregation methods besides averaging and determine under which conditions an aggregation method is better than others.

In this paper, we provide a first approximation of the aggregation of learning models in the FL paradigm using neural networks with different learning processes (error-based and memory-based). Moreover, we develop a federated learning system to simulate the clients through an on-device learning process, and the central entity that receives the models to start the aggregation process. Specifically, we formulate a scenario like the FedAvg three-stage structure; however, the main goal is to explore how the aggregation process works in different neural networks and compare them under different configurations through performance metrics.

Attending to the arguments, this paper gives an analysis to understand the aggregation process, explore the model aggregation techniques, and show how they can be integrated into neural networks with different learning processes. The rest of the paper is organized as follows. Section 3 presents the related work. Section 4 presents the system model and implementation of the FL system. Section 6 presents the results of some experiments between the different aggregation types. Section 7 details the envisioned contributions and concludes the paper.

## 2 Preliminaries

In section 2.1, we present a brief description of key concepts related to federated learning, including the deployment strategies (cloud, edge, hybrid) and learning approaches (horizontal and vertical). Then, section 2.2 presents an overview of neural networks, including the most commonly used learning processes (error-based and memory-based).

**Table 1.** Model aggregation techniques.

| Reference | Aggregation technique | Description |
|---|---|---|
| [2, 21] | Complete<br>Averaging (CA) | The central entity averages local models of all participating clients. |
| [21] | Best Half<br>Averaging (BHA) | The central entity averages local models of the best half of participating clients. |
| [21] | Best Model<br>Selection (BMS) | After receiving all local models, the central entity chooses the best model. |

### 2.1 Overview of Federated Learning

Federated learning is conceived of as a new distributed learning paradigm that has received a lot of interest lately since it addresses some of the privacy and data security concerns identified in the conventional paradigm. A basic federated learning system consists of a group of clients who train a local model based on their local data, upload the model to a central entity, and aggregate the models of each user. Figure 1 depicts a straightforward federated learning system.

Deployment techniques and learning approaches are two crucial elements that must be integrated into a federated learning system. To create a global model in FL, computing entities deployed at a facility for edge computing or cloud computing could potentially be used. Figure 2 illustrates the three deployment options for an FL system, which can be classified as cloud, edge, and hybrid.

Figure 2a shows the cloud-based FL deployment strategy, where each device creates a unique model through a training process with its data; the parameters of these models are then updated on the cloud to produce an aggregated model. Figure 2b shows that the base station's cell site's edge computing nodes will combine the base stations' separate learning models. This form of architecture is suitable from the standpoint of communication to satisfy the latency requirements related to updating the individual models toward the computing unit in charge of the aggregate.

A set of users within a specific coverage region can also acquire more representative global learning models thanks to the FL at the edge. Eventually, you can benefit from both cloud and edge computing by using the hybrid strategy presented in Figure 2c. In other words, a local model aggregation entity at the edge receives model changes from mobile devices quickly. Then, to create a global model, models that were uploaded to the edge nodes are sent to the cloud, where an aggregation process is carried out.

On the other hand, if the data properties of each client are consistent, an FL global model can produce the same modeling results as a centralized paradigm [3]. Three basic categories can be used to categorize federated learning based on the features of the data from the participating clients [27]. Each category is briefly detailed below.

- **Horizontal FL**: Also known as sample-based federated learning, this category is used when datasets share the same feature space but different sample sizes.
- **Vertical FL**: Also referred to as feature-based federated learning, this category can be used when two datasets have different feature spaces but the same sample identification space.

**Table 2.** Summary of related work.

| Reference | FL Algorithm | Deployment Architectures | | | Machine Learning Algorithms | | | | | Experiments Information | | Model Aggregation Type |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | Memory-based | Error-based | | | | | | |
| | | Cloud | Edge | Hybrid | RBFN | CNN | MLP | LIR | LOR | Dataset | Clients | |
| This work, 2023 | FedRBFN | ✓ | - | - | ✓ | - | ✓ | ✓ | - | Synthetic | 5-50 | CA |
| | | | | | | | | | | | | BHA |
| | | | | | | | | | | | | BMS |
| [20], 2022 | FedBCD | ✓ | - | - | - | ✓ | - | - | ✓ | MIMIC III | - | CA |
| | | | | | | | | | | MNIST | | |
| | | | | | | | | | | NUS-WIDE | | |
| | | | | | | | | | | Default-Credit | | |
| [26], 2022 | FedHome | - | - | ✓ | - | ✓ | ✓ | - | - | MobiAct | 57 | CA |
| [14], 2021 | SCAFFOLD | ✓ | - | - | - | - | ✓ | - | - | MNIST | 100 | CA |
| [6], 2020 | FedHealth | ✓ | - | - | - | ✓ | ✓ | - | - | UCI Smartphone | 5 | CA |
| | | | | | | | | | | PCAP | 20 | |
| | | | | | | | | | | COVID-19 | 20 | |
| [18], 2020 | FedProx | ✓ | - | - | - | - | - | - | ✓ | MNIST | 1000 | CA |
| | | | | | | | | | | FEMNIST | 200 | |
| | | | | | | | | | | SHAKESPEARE | 143 | |
| | | | | | | | | | | SENTIMENT140 | 772 | |
| [19], 2020 | FedGRU | ✓ | - | - | - | - | ✓ | - | - | PeMS | 20 | CA |
| [25], 2020 | FedMA | - | ✓ | - | - | ✓ | - | - | - | CIFAR-10 | 16 | CA |
| | | | | | | | | | | SHAKESPEARE | 66 | |
| [23], 2020 | FedPAQ | - | ✓ | - | - | - | ✓ | - | ✓ | CIFAR-10 | 50 | CA |
| | | | | | | | | | | MNIST | | |
| [12], 2020 | FedAttOpt | ✓ | - | - | - | - | ✓ | - | - | Penn Treebank | 100 | CA |
| | | | | | | | | | | WikiText-2 | | |
| | | | | | | | | | | Reddit Comments | | |
| [22], 2019 | FedCS | - | ✓ | - | - | ✓ | - | - | - | CIFAR-10 | 100 | CA |
| | | | | | | | | | | Fashion MNIST | 1000 | |
| [17], 2019 | FedMD | ✓ | - | - | - | ✓ | - | - | - | MNIST | 10 | CA |
| | | | | | | | | | | FEMNIST | | |
| | | | | | | | | | | CIFAR-10 | | |
| | | | | | | | | | | CIFAR-1000 | | |
| [1], 2019 | FedPer | ✓ | - | - | - | ✓ | - | - | - | CIFAR-10 | 10 | CA |
| | | | | | | | | | | CIFAR-100 | 10 | |
| | | | | | | | | | | FLICKR-AES | 30 | |
| [2], 2017 | FedAvg | ✓ | - | - | - | ✓ | - | - | - | CIFAR-10 | 100 | CA |
| | | | | | | | | | | MNIST | 100 | |
| | | | | | | | | | | SHAKESPEARE | 1146 | |

RBFN = Radial Basis Function Networks, MLP = Multi-Layer Perceptron, CNN = Convolutional Neural Networks, NN = Neural Networks

## 2.2 Overview of Neural Networks

Neural networks can learn from their surroundings and enhance their performance through learning. A neural network learns through an interactive process of adjusting synaptic weights and bias levels, and over time, performance improves by some predetermined metric. After each iteration of the learning process, the networks should ideally become more informed about their environment. A few learning methods that are useful in the context of machine learning are: 1) error-based learning, and 2) memory-based learning [9].

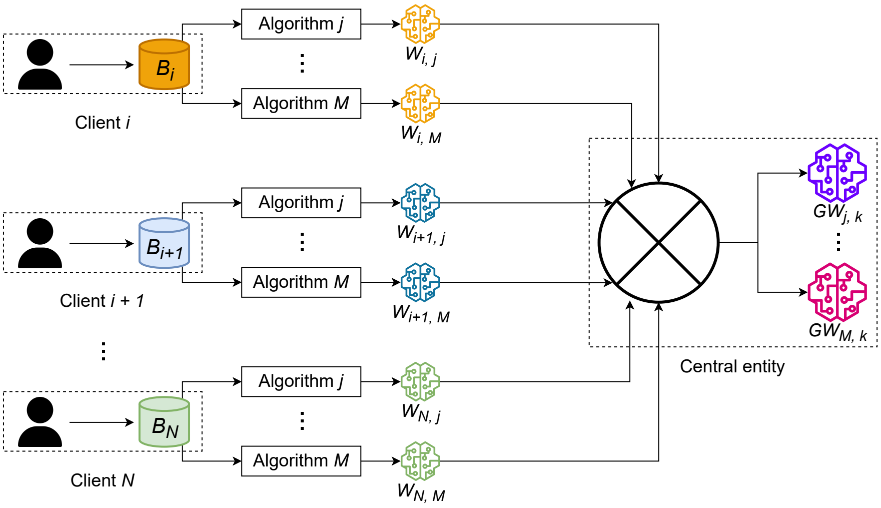


**Fig. 3.** System model.

In error-based learning, models adjust their parameters based on the discrepancy between their predictions and the true target values in the training data. This is achieved by optimizing a chosen loss function, which measures the error, using techniques like backpropagation. Error-based learning models that are good examples include multilayer perceptron (MLP) and linear regression (LR).

MLP, a feedforward neural network, learns complex non-linear relationships between input features and target labels through hidden layers. MLPs use backpropagation, a popular error-based learning algorithm, to update the network's weights and minimize the error between predicted and target outputs during training. Meanwhile, LR aims to predict the probability of a binary outcome by adjusting its parameters through gradient descent to minimize the error between predicted and actual target values. Both models iteratively refine their parameters to minimize prediction errors and make accurate generalizations about unseen data.

On the other hand, memory-based learning models, such as radial basis function (RBF) neural networks, store training examples explicitly in memory and make predictions based on their similarity to new inputs. RBF networks use radial basis functions to represent the data and find the closest training examples to new inputs during prediction [16]. This memory-based approach allows RBF networks to perform well in interpolation and pattern recognition tasks. Unlike error-based models, RBF does not explicitly minimize a predefined error function but instead retrieves relevant training instances from memory to make predictions.

## 3 Related Work

In this section, we discuss some federated learning algorithms proposed in the literature that address different concerns and the integration of use cases in several areas such as healthcare, security, business, and industry.

**Algorithm 1** On-device learning pseudocode.

```
Require: Data batch $B_i$.
Ensure: All local models with structure $W_{i,j}$.
for $j$ in $M$ do
    for $i$ in $N$ do
        Split $B_i$ into training and test data.
        Initialize an instance of algorithm $j$, specifying the parameters.
        Start the training process of the model with the training data.
        Validate through a predicted task with test data.
        Get model performance metric.
        Save local model.
        Transmit local model $W_{i,\,j}$ to central entity.
    end for
end for
```

As we have mentioned before, most of the proposed solutions are based on the FedAvg algorithm, implementing the three-stage structure, the same aggregation method, and varying the deployment technique as well as the learning approach.

### 3.1 Aggregation Methods

Aggregation methods in federated learning are essential because of their role in updating global models, whether the messages exchanged are the models themselves, some or all of their parameters, or gradients. However, aggregation in federated machine learning goes beyond simply merging model updates. In addition, aggregation can be carried out hierarchically, aggregating local models on intermediate servers before sending them to the central server, enabling large-scale federated learning systems.

This is why the aggregation algorithm is such a fundamental concept in federated learning; it ultimately determines the success of model training and whether or not the resulting model is practical to use. Some of the aggregation methods are mentioned in Table 1, since there may be approaches other than those presented here.

In federated learning, a variety of aggregation algorithms are used depending on the goals to be achieved, such as protecting user privacy, increasing the convergence rate, and reducing the damage caused by fraudulent customers. Each of these approaches has its advantages and disadvantages, and some are better suited to certain contexts of federated learning than others.

### 3.2 Federated Learning Algorithms

In federated machine learning, the central entity and clients work together to train a global model. Numerous aggregation techniques have been developed in recent years as a result of the various methods used to aggregate locally trained models. Federated learning became a popular topic that attracted scholars and inspired thousands of experiments in other areas. Table 2 highlights important factors, including the machine learning algorithms utilized, client selection strategies, and model aggregation type. For instance, McMahan et al. [2] proposed a practical approach to federated learning of deep networks that uses an iterative model averaging technique.

**Algorithm 2** Model aggregation pseudocode.

```
Require: All local models with structure $W_{i,j}$.
Ensure: Global model $GW_{j,k}$.
Receive all local models from participant clients.
for each algorithm $j \in M$ do
    Aggregate models by complete averaging.
    Aggregate models by best half models.
    Aggregate models by best model selection.
    Evaluate and compare all aggregations.
    Save global model $GW_{j,k}$.
end for
```

In this approach, they carry out an empirical evaluation considering four datasets. The results of the experimentation show that the FedAvg approach trains high-quality models with relatively few rounds of communication. This can be seen in the results from a variety of model architectures, such as a multilayer perceptron and two different convolutional neural networks. Hard et al. [8] proposed an algorithm where a recurrent neural network linguistic model was trained through a distributed learning framework in the device to predict the next word on a virtual keyboard on smartphones.

In addition, a comparison was made between the local models and the global model. In particular, the authors demonstrate the feasibility and benefits of training linguistic models on devices without exporting sensitive user data to the central node. On the other hand, Zhu et al. [28] discuss the development of a non-segmented text recognition model based on a neural network for text recognition of a corpus of Chinese financial documents, which often contain personal, confidential, and critical information.

The comparison of the results showed that the use of an FL framework effectively improves the aggregate performance of the text recognition model without sharing sensitive data. In the healthcare context, some FL approaches have been proposed. Kaissis et al. [13] presented a research paper detailing the benefits of using FL algorithms to improve patient care and, above all, to simultaneously address demands for the protection and use of data.

Meanwhile, Sadilek et al. [24] work demonstrates that creating federated learning models can achieve similar accuracy, precision, and generalizability, and lead to the same interpretation as centralized models while achieving significantly greater privacy protection without significantly increasing computational costs. To do this, they use a diverse set of health studies from one and several health centers.

Nevertheless, federated learning can be applied to contexts related to finance, business, and industry. Hiessl et al. [10] introduced an FL-based industrial system that supports knowledge sharing in the evaluation and continuous updating of learning tasks with sufficient data similarity to enable optimal collaboration of trading partners on common ML problems.

It prevents negative knowledge transfer and ensures resource optimization of the edge devices involved. Kawa et al. [15] developed a method based on federated learning that allows institutions to collaboratively learn a shared prediction model, keeping all the data in the banking institutions. In this way, the ability to do machine learning would be decoupled from the need to store data in the cloud, thus eliminating centralization.

**Table 3.** Simulation parameter settings.

| **Parameters** | **Value** | | | |
|---|---|---|---|---|
| | **System** | **RBFN** | **MLP** | **LR** |
| Number of clients | [10, 20, 30, 40, 50] | - | - | - |
| Number of samples | 125000 | - | - | - |
| Number of clusters | - | 8 | - | - |
| Gamma | - | 0.0009 | - | - |
| Beta | - | - | 0.05 | - |
| Number of iterations | - | - | 300 | 1000 |
| Learning rate | - | - | 0.01 | 0.01 |

Jansson [11] proposed a framework for training a federated learning fraud detection model in which multiple entities collaborate to solve an ML problem under the coordination of a central server. In this way, financial institutions can collectively reap the benefits of a shared model, which has seen more fraud than each bank alone, without sharing the data set among themselves.

## 4 System Model

The system modeling is illustrated in Fig. 3. We consider a generic federated learning system with $N$ clients denoted as $i \in \{1, 2, \ldots, N\}$; and $M$ algorithms denoted as $j \in \{1, 2, \ldots, M\}$. Each client $i$ has a data batch denoted as $B_i$ with all the data gathered over a certain amount of time. Additionally, $B_i$ serves as the input data for each algorithm $j$, which results in the learning model $W_{i,\,j}$.

Each participant client $i$ is assumed to be connected to a cloud facility that provides computing and storage resources. The cloud server aggregates models with $W_{i,\,j}$ structure during each learning round to generate a global model denoted as $GW_{j,\,k}$. Finally, it is important to emphasize that the structure of our FL system is based on the FedAvg with a few modifications.

For instance, no selection criteria were considered, and all clients were eligible to participate in the aggregation process. In addition, transmission rates and backhaul resources are not considered because our analysis is focused on understanding the aggregation methods under different scenarios.

Additionally, we consider the aggregation techniques described in Table 1 because, as we have mentioned before, averaging methods are the most understandable and commonly used methods to aggregate learning models. The CA method is considered the standard method for aggregation since FedAvg, and all its variants have integrated it into their model aggregation process because it ensures that the learning models of all clients are averaged to generate a generalized model. In this sense, the global model generation through CA is expressed as:

$$GW_{j,k} = CA\left(\sum_{i=1}^{N} W_{i,\,j}\right) = \frac{W_{1,\,j} + W_{2,\,j} + \ldots + W_{i,\,j}}{N}. \tag{1}$$

Similar to this, the BHA approach takes into consideration the top half of the client-submitted models. A performance test of the models is used to inform the decision-making process. For instance, let's assume four clients create and evaluate a model, which is later transmitted to the central entity. In this case, the central entity decides through BHA which of all models is going to be aggregated. Equation 2 shows the global model generated in BHA using the same example:

$$GW_{j,k} = \mathrm{BHA}\left(\sum_{i=1}^{N=4} W_{i,\,j}\right) = \frac{W_{1,j} + W_{4,j}}{\frac{N}{2}}. \tag{2}$$

The BMS technique, which differs from the average and involves choosing the best model from the clients, can be used to simplify the aggregation process so that it is easier to grasp. Returning to the previous example in this sense, the difference is that the central entity will now only choose the best model out of all of them. The global model generated by the BMS aggregation can be expressed as follows:

$$GW_{j,k} = \mathrm{BMS}\left(\sum_{i=1}^{N=4} W_{i,\,j}\right) = W_{1,j}. \tag{3}$$

## 5 Proposed Approach

Our suggested federated learning system follows the conventional three-phase structure of all federated learning algorithms; however, we omit the participant client selection stage because we assume that clients have already been previously selected through a selection technique. In this regard, the two remaining phases of our approach are on-device training and model aggregation, which are detailed below.

### 5.1 On-Device Learning

At this stage, participant clients have already been selected and modeled as smart devices (smartphones). It is also assumed that the devices are distributed in a coverage area with good computational capacity and communication channel conditions, where each smartphone has enough power capacity to carry out a training process. In federating learning systems, modeling a device with a data collection process is commonly considered. For this, each device already possesses a data batch with data collected over a period of time. Each data batch acts as input for any learning algorithms the device desires to employ.

In this situation, neural networks like RBFN, MLP, and LR have been taken into consideration because the goal of our approach is to solve a problem involving linear regression and improve the results' interpretability. Once the training process ends, the output for each algorithm is a learning model, which has its own structure. Finally, the validation step involves taking the test data and calculating its performance in terms of the coefficient of determination. In Algorithm 1, the main steps of the on-device learning stage are presented.

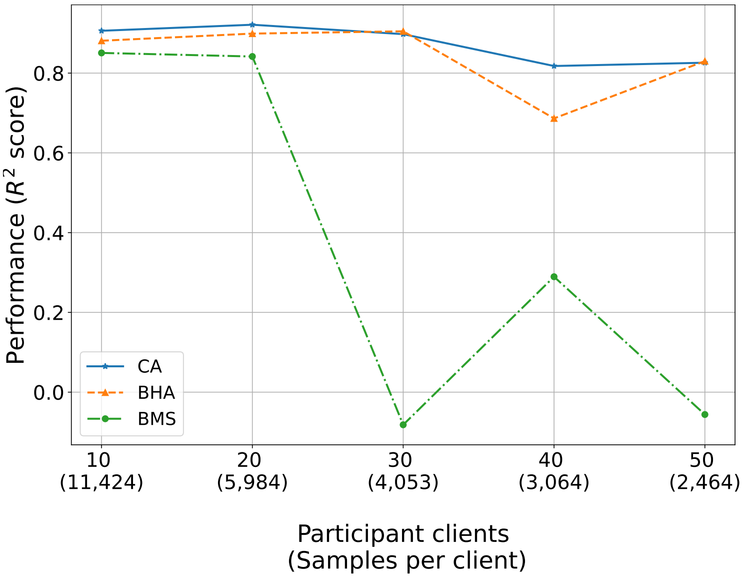


**Fig. 4.** RBFN performance varying the number of clients from 10 to 50.

### 5.2 Model Aggregation

On the other hand, Algorithm 2 shows the steps that make up our approach running on the central entity side. Initially, the central entity is modeled as a cloud server with computational and storage capacity because it is assumed that our federated learning system will be based in the cloud. The main task of the cloud server is receiving the local models from participating clients. The models are combined to generate a new generalized model using an aggregation method after all local models have been transmitted to the cloud server.

Complete averaging, best half averaging, and best model selection are regarded as aggregation methods in this work since they tend to be excellent representations of aggregation methods in federated learning. Once the unification process is complete, each algorithm $j$ produces a global learning model that keeps the original local models' structural integrity. A validation procedure is then used to assess the new model's performance and determine whether the coefficient of determination has improved. All these steps are repeated $k$ learning rounds.

## 6 Experiments

In this section, the effectiveness of the suggested federated learning system is determined using in-depth simulations. To demonstrate that model averaging isn't the best option in federated learning methods, the simulation settings will be shown first, followed by the simulation results.

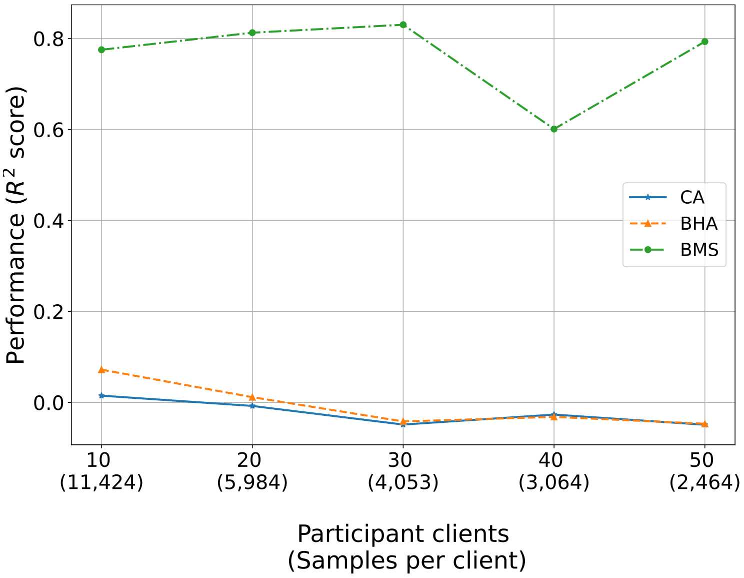


**Fig. 5.** MLP performance varying the number of clients from 10 to 50.

### 6.1 Simulation Setup

We simulate a common cloud-based federated learning system with smart devices dispersed throughout a coverage region. We chose a homogenous setting for this simulation where each device has a data batch with the same number of samples, creating the same conditions for each device to perform the training process. Additionally, we looked at a linear regression problem where the effectiveness of the aggregation process is assessed using complete averaging, best half averaging, and best model selection techniques.

We also contrast these aggregating methods across three distinct neural network designs. The first two relate to methodologies frequently utilized in error-based learning, while the third strategy makes use of memory-based learning. Table 3 shows the parameters used for the system configuration, including the neural network parameters. Specifically, the following provides information about the examined algorithms:

- Multilayer Perceptron (MLP). It is a versatile algorithm that can rupture complex relationships in data due to its multiple hidden layers and nonlinear activation functions.
- Linear Regression (LR). It is straightforward to interpret, making it a great choice for cases where the relationship between variables seems linear.
- Radial Basis Function Network (RBFN). It focuses on localized patterns within the data using radial basis functions, which allows it to capture different behaviors in different regions of the input space.

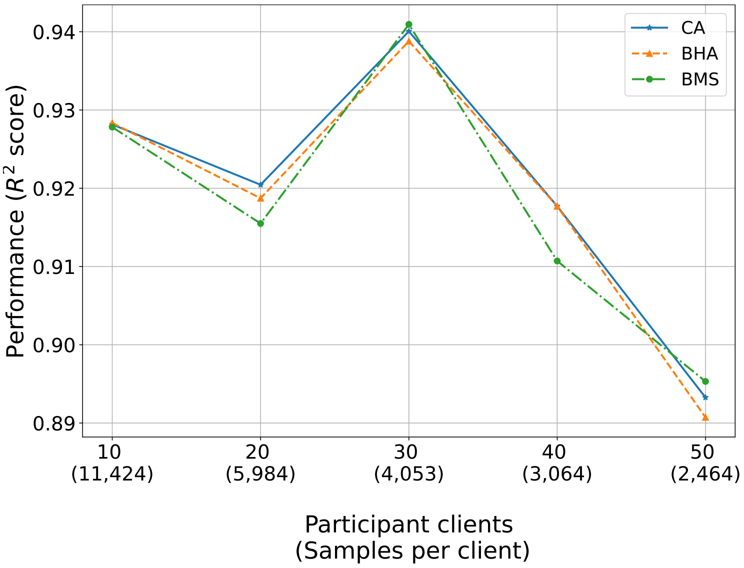


**Fig. 6.** LR performance varying the number of clients from 10 to 50.

### 6.2 Metrics

For each learning round in the system, a set of clients transmits local models to a central entity, which is in charge of the aggregation process. However, for each generated model, the following performance metrics are obtained:

- Coefficient of Determination ($R^2$ Score). It represents the proportion of variance in the dependent variable that is predictable from the independent variables in the model. In other words, it measures the goodness of fit of the model. In addition, it is easy to interpret. A value of 1.0 indicates that the model perfectly predicts the variations in the dependent variable, while a value of 0.0 suggests that the model's predictions do not explain any variability beyond the mean of the dependent variable.

### 6.3 Performance Analysis

The effectiveness of RBFN in a federated learning system is displayed in Figure 4. It demonstrates a comparison of the three model aggregating methods for five different configurations in more detail. In this experiment, we adjusted both the total number of clients and the average number of samples per client so that, while the total number of clients rose, the average number of samples fell.

Results of the experiment reveal that BHA and CA perform similarly across all client configurations, while BMS performs reasonably well for 10 and 20 clients. It is critical to emphasize that BMS suffers when the number of clients grows since it becomes more challenging to maintain a generalized model with only one client's knowledge. On the other hand, Figure 5 shows the performance of MLP, which suggests that averaging methods from model aggregation, such as CA and BHA, are not the best for this architecture. The average strategy performs poorly for the configuration of 10 users, with a $r^2$ score of less than 0.2. The experiment continues with this behavior in the subsequent setups, where it even appears to get worse as the number of clients rises. Instead, the aggregation strategy for the best model selection offers a good performance even when the number of users is increasing.

For all configurations, BMS keeps its $r^2$ score value above 0.6; the setup with 30 clients is the only one that scores higher than 0.8. The performance of the LR algorithm in various federated learning system setups is shown in Figure 6. It demonstrates that under various configurations of client and sample numbers, the behavior of the three model aggregation strategies is nearly identical. But it is crucial to note that, except in the trial with 30 participant clients, where CA, BHA, and BMS are all above 0.93 and only the BMS exceeds the 0.94 target, the performance of LR decreases in terms of $r^2$ score as the number of clients increases.

## 7 Conclusions

In this paper, we have studied the aggregation process through different techniques in non-common scenarios. Furthermore, we discuss the importance of model aggregation to generate a generalized global model that represents all users involved in collaborative training. Simulation results have shown that the CA technique is not always the most effective way to aggregate learning models, even when it is mostly used in FedAvg algorithm variants.

However, other model averaging techniques such as BHA and BMS should be considered in more federating learning scenarios because, in some cases, they have better performance than the CA technique. In future work, we plan to integrate the selection stage, which is a fundamental part of determining the participant users, and model a communication system that allows us to play scenarios with different deployment approaches such as cloud, edge, and hybrid.

## References

1. Arivazhagan, M. G., Aggarwal, V., Singh, A. K., Choudhary, S.: Federated learning with personalization layers (2019) doi: 10.48550/ARXIV.1912.00818
2. Brendan-McMahan, H., Moore, E., Ramage, D., Hampson, S., Aguera-y-Arcas, B.: Communication-efficient learning of deep networks from decentralized data. In: Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, vol. 54 (2016) doi: 10.48550/ARXIV.1602.05629
3. Chai, Z., Fayyaz, H., Fayyaz, Z., Anwar, A., Zhou, Y., Baracaldo, N., Ludwig, H., Cheng, Y.: Towards taming the resource and data heterogeneity in federated learning. In: USENIX Conference on Operational Machine Learning, pp. 19–21 (2019)
4. Chen, M., Poor, H. V., Saad, W., Cui, S.: Convergence time optimization for federated learning over wireless networks. IEEE Transactions on Wireless Communications, vol. 20, no. 4, pp. 2457–2471 (2021) doi: 10.1109/twc.2020.3042530
5. Chen, M., Yang, Z., Saad, W., Yin, C., Poor, H. V., Cui, S.: A joint learning and communications framework for federated learning over wireless networks. IEEE Transactions on Wireless Communications, vol. 20, no. 1, pp. 269–283 (2021) doi: 10.1109/twc.2020.3024629

6. Chen, Y., Qin, X., Wang, J., Yu, C., Gao, W.: Fedhealth: A federated transfer learning framework for wearable healthcare. IEEE Intelligent Systems, vol. 35, no. 4, pp. 83–93 (2020) doi: 10.1109/mis.2020.2988604
7. Ferdowsi, A., Saad, W.: Generative adversarial networks for distributed intrusion detection in the internet of things. In: IEEE Global Communications Conference, pp. 1–6 (2019) doi: 10.1109/globecom38437.2019.9014102
8. Hard, A., Rao, K., Mathews, R., Ramaswamy, S., Beaufays, F., Augenstein, S., Eichner, H., Kiddon, C., Ramage, D.: Federated learning for mobile keyboard prediction. arXiv (2018) doi: 10.48550/ARXIV.1811.03604
9. Haykin, S.: Neural networks: A comprehensive foundation. Prentice Hall PTR (1998)
10. Hiessl, T., Schall, D., Kemnitz, J., Schulte, S.: Industrial federated learning – requirements and system design. In: Highlights in Practical Applications of Agents, Multi-Agent Systems, and Trust-worthiness. The PAAMS Collection, vol. 1233, pp. 42–53 (2020) doi: 10.1007/978-3-030-51999-5_4
11. Jansson, M., Axelsson, M.: Federated learning used to detect credit card fraud. Master's Thesis, Department of Computer Science, LUND University (2020)
12. Jiang, J., Ji, S., Long, G.: Decentralized knowledge acquisition for mobile internet applications. World Wide Web, vol. 23, no. 5, pp. 2653–2669 (2020) doi: 10.1007/s11280-019-00775-w
13. Kaissis, G. A., Makowski, M. R., Rückert, D., Braren, R. F.: Secure, privacy-preserving and federated machine learning in medical imaging. Nature Machine Intelligence, vol. 2, no. 6, pp. 305–311 (2020) doi: 10.1038/s42256-020-0186-1
14. Karimireddy, S. P., Kale, S., Mohri, M., Reddi, S., Stich, S., Suresh, A. T.: Scaffold: Stochastic controlled averaging for federated learning. In: Proceedings of the 37th International Conference on Machine Learning, vol. 119, pp. 5132–5143 (2020)
15. Kawa, D., Punyani, S., Nayak, P., Karkera, A., Jyotinagar, V.: Credit risk assessment from combined bank records using federated learning. International Research Journal of Engineering and Technology, vol. 6, no. 4, pp. 1355–1358 (2019)
16. Kriesel, D.: A brief introduction to neural networks (2007) www.dkriesel.com
17. Li, D., Wang, J.: FedMD: heterogenous federated learning via model distillation. In: Proceedings of the 33rd Conference on Neural Information Processing Systems (2019) doi: 10.48550/ARXIV.1910.03581
18. Li, T., Sahu, A. K., Zaheer, M., Sanjabi, M., Talwalkar, A., Smith, V.: Federated optimization in heterogeneous networks. In: Proceedings of the 1st Adaptive and Multitask Learning Workshop (2018) doi: 10.48550/ARXIV.1812.06127
19. Liu, Y., Yu, J. J. Q., Kang, J., Niyato, D., Zhang, S.: Privacy-preserving traffic flow prediction: a federated learning approach. IEEE Internet of Things Journal, vol. 7, no. 8, pp. 7751–7763 (2020) doi: 10.1109/jiot.2020.2991401
20. Liu, Y., Zhang, X., Kang, Y., Li, L., Chen, T., Hong, M., Yang, Q.: Fedbcd: a communication-efficient collaborative learning framework for distributed features. IEEE Transactions on Signal Processing, vol. 70, pp. 4277–4290 (2022) doi: 10.1109/tsp.2022.3198176
21. Mahedi-Hasan, H. M.: Analysis of model aggregation techniques in federated learning. Master's Thesis, Faculty of Graduate Studies and Research, University of Regina (2021)
22. Nishio, T., Yonetani, R.: Client selection for federated learning with heterogeneous resources in mobile edge. In: IEEE International Conference on Communications, pp. 1–7 (2019) doi: 10.1109/icc.2019.8761315
23. Reisizadeh, A., Mokhtari, A., Hassani, H., Jadbabaie, A., Pedarsani, R.: FedPAQ: A communication-efficient federated learning method with periodic averaging and quantization. Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics (2020) doi: 10.48550/ARXIV.1909.13014

24. Sadilek, A., Liu, L., Nguyen, D., Kamruzzaman, M., Serghiou, S., Rader, B., Ingerman, A., Mellem, S., Kairouz, P., Nsoesie, E. O., MacFarlane, J., Vullikanti, A., Marathe, M., Eastham, P., Brownstein, J. S., Aguera-y-Arcas, B., Howell, M. D., Hernandez, J.: Privacy-first health research with federated learning. npj Digital Medicine, vol. 4, no. 1 (2021) doi: 10.1038/s41746-021-00489-2
25. Wang, H., Yurochkin, M., Sun, Y., Papailiopoulos, D., Khazaeni, Y.: Federated learning with matched averaging. In: Proceedings of the 8th International Conference on Learning Representations (2020) doi: 10.48550/ARXIV.2002.06440
26. Wu, Q., Chen, X., Zhou, Z., Zhang, J.: Fedhome: cloud-edge based personalized federated learning for in-home health monitoring. IEEE Transactions on Mobile Computing, vol. 21, no. 8, pp. 2818–2832 (2022) doi: 10.1109/tmc.2020.3045266
27. Yang, Q., Liu, Y., Chen, T., Tong, Y.: Federated machine learning: concept and applications. ACM Transactions on Intelligent Systems and Technology, vol. 10, no. 2, pp. 1–19 (2019) doi: 10.1145/3298981
28. Zhu, X., Wang, J., Hong, Z., Xia, T., Xiao, J.: Federated learning of unsegmented chinese text recognition model. In: Proceedings of the IEEE 31st International Conference on Tools with Artificial Intelligence, pp. 1341–1345 (2019) doi: 10.1109/ictai.2019.00186

ISSN 1870-4069

# Front-End Responsive Web Application for Chatbot Retraining Using ReactJS with MongoDB

Luis A. Fernández-Castro[1], Javier A. Valdivia-del-Ángel[1],
Kevin G. Zazueta-Sánchez[2], Ricardo Cantú-Martínez[2],
Yoel Ledo-Mezquita[1], Elioth Macias-Frotto[1]

[1] Tecnologico de Monterrey,
School of Engineering and Sciences,
Mexico

[2] NDS Cognitive Labs. Prolongación,
Mexico

{a01023675, a01658296, yledo, a00837025}@tec.mx,
{kzazueta, rcantu}@ndscognitivelabs.com

**Abstract.** This article shows the front-end implementation of a responsive web application for chatbot retraining, and the design justification for it.

**Keywords:** Front-end, responsive web application, chatbot retraining, reactJS, MongoDB.

## 1 Introduction

It was during the 1960s that computer science professor Joseph Wiezenbaum, at the artificial intelligence laboratory of the Massachusetts Institute of Technology, created the first chatterbot, later shortened to a chatbot, called "Eliza" [1]. The program sought to emulate a psychotherapist by recognizing patterns and interacting with the user through the use of a pre-set script.

Currently, chatbots are software or computer programs that simulate human conversation or "chatter" through text messages or voice interactions [2]. In addition, these programs are much more robust than their predecessors, and increasingly resemble communicating with another human being.

The importance of chatbots has been increasing because they increase efficiency and productivity, since these automated programs are delegated with tasks mainly of sales, customer service or virtual assistance. But above all, while consumers are moving away from traditional forms of communication, more and more chat-based communication methods are expected [3].

As chatbots have become more complex in their comprehension and communication capabilities, the way these programs are created, trained, and retrained has also evolved in order to improve their functionalities in accordance with their repeated interaction with users. In this sense, chatbots have gone from using predefined scripts to using machine learning algorithms to adapt as they interact more with the user.

pp. 137–142; rec. 2023-06-04; acc. 2023-09-01

One of the most important phases in the life cycle of a chatbot is its retraining, since it has been able to go through previous stages of design and construction where its programmer or programmers have made a judgment of its optimal behavior, but by using real user data, it is possible to evaluate whether these prejudices have been correct or not and the chatbot can be adapted to the true needs of the chatbot. user.

The intention of this article, beyond restricting itself to treating chatbots and their algorithms as its main thesis, seeks to recognize the chatbot retraining phase as a central point of study, more specifically the semi-automation of this from another computer program, which serves as an interface.

This framework includes the delimitation of the retraining phase itself and the technologies used to carry out this program. Therefore, we will delve into the technologies used and their justification for this, and we will underline the importance of these with examples of use of the product that is being developed in conjunction with NDS Cognitive Labs for this same task.

## 2 Chatbot Retraining

To highlight the issue of chatbot retraining, it is necessary to think about the reason for the success of these computer programs. This software depends on their adaptability and problem solving, and this is why it is possible that when they cannot adapt to the needs of the user, they must at least have the ability to retrain [4].

### 2.1 Manual Chatbot Retraining

The manual retraining of chatbots is simply known as conventional training, and it is very similar to the training that is done before the publication of this system, with the difference that now there is additional information, hundreds of data about their interaction with users, which helps the programmer or programmers to review the weak points of the program.

Depending on the complexity of the computer program being discussed, whether it is a chatbot based on keyword recognition, one based on menus of options, or a hybrid that combines the two mentioned above, is how the training is going to be done, either including more options that allude to the needs of the consumer or expanding the list of keywords and the answers needed for them.

### 2.2 Semi-Automatic Chatbot Retraining

It is called semi-automatic retraining because despite the advances that have been made in technology in recent years, it has not yet been possible to create a program that learns completely on its own through its interaction with users and does not depend on at least one programmer in the evaluation and retraining phase. However, the goal of this automation is to rely on a minimum of one person for the program to adapt, and not to require fundamental changes to its base algorithm.

Retraining is simplified as much as possible and programming structures are used at its base so that the information remains consistent throughout this retraining phase.

**Table 1.** Comparison of manual and semi-automatic retraining.

| **Manual** | **Semi-automatic** |
|---|---|
| Difficult to maintain | Easy to maintain |
| Tighter monitoring of chatbot interactions | No code modification required |
| Requires code modifications | Reduced retraining times |
| High retraining times | Use a graphical interface |
| Changes are made to the data or code | |

### 2.3 Is Manual or Semi-Automatic Retraining Better?

While both retraining's require one or more programmers to be evaluated and readapted, it certainly requires less effort to maintain semi-automatic retraining. See Table 1. In addition, because manual retraining is usually very similar to training in design and construction stages, it will often require modifying the code or modifying the data from its containers.

One of the biggest advantages of semi-automatic retraining is the use of a graphical interface, in this way you can visualize the data and perform different actions on it, such as creating, reading, editing, and deleting it. In addition, a better data flow can be ensured, ensuring that the information is complete regardless of the data entered by the person in charge of the retraining.

## 3 Responsive Web App

Web applications "are programs that work on the Internet. In other words, the data or files you work on are processed and stored within the website. These applications usually don't need to be installed on your computer." GCFGlobal [5]. This framework includes the front-end and back-end parts, which comprise the application as a whole, but take care of different processes.

### 3.1 Back-End in a Web Application

The back-end comprises the data and logic part in a web application, and is hidden from the user [5].

Different frameworks or technologies can be used to handle this part, but typically two technologies are used together, one for data handling, and one for logic and functional structure.

### 3.2 Front-End in a Web Application

The front-end comprises the user-interactive parts, which are displayed in the browser, and is visible to all users [5].

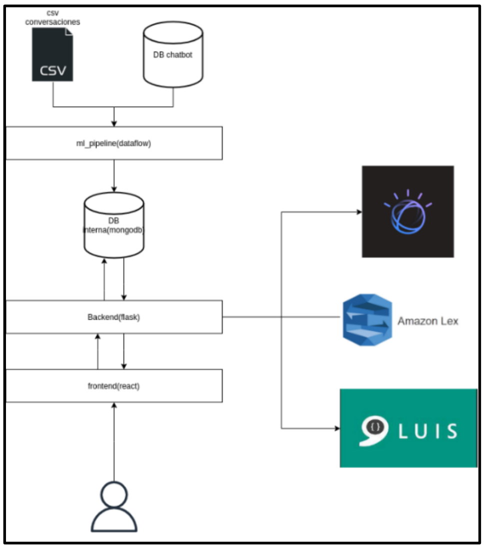


**Fig. 1.** Web Application Proposal Design Scheme (2022).

The front-end is one of the parts with the largest number of frameworks and options to work on it. This is due to the fact that the visual or 'interface' part is mainly worked on.

### 3.3 Web Application Proposal

Guided by Kevin G. Zazueta and Ricardo Cantú, engineers at NDS Cognitive Labs, it has been possible to develop a proposal for a web application for the retraining of chatbots, see Fig. 1.

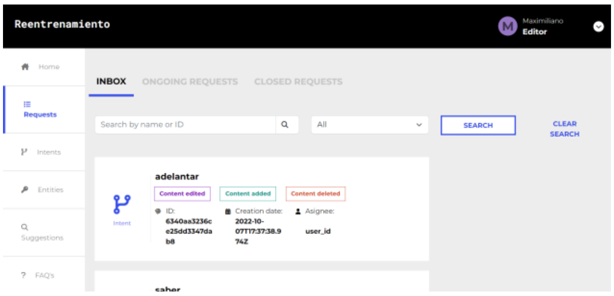
For this implementation, the use of the Flask framework is proposed to handle the back-end, and React for the front-end, while the data will be managed with MongoDB.

The main objective of this development is to be a robust system that can retrain several different chatbots and is easy to use for the user.

### 3.4 Why a responsive Web App?

As the internet has evolved, it is increasingly used on mobile devices than on the computers where it was originally used. This has been a motivation to adapt as many programs as possible to versions that can be seen on screens of different resolutions.

While it is true that it is more effective to take advantage of a larger interface for a program with the complexity proposed, limitations must be considered with larger screens, for example, that a user has to travel and during the journey it is easier to continue with their work from the mobile device. This is why the responsive part must be considered, that is, depending on the size of the screen resolution, it adapts to it.



adelantar

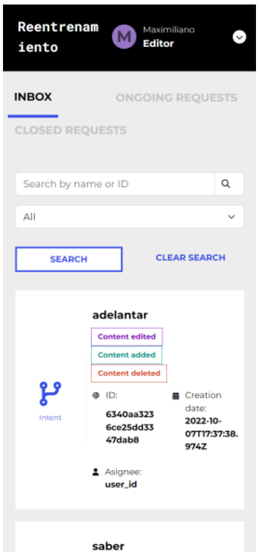
ID: 6340aa3236ce25dd3347dab8

Creation date: 2022-10-07T17:37:38.974Z

Asignee: user_id

saber

**Fig. 2.** Requests page view in chatbot retraining web app (2022).



adelantar

ID: 6340aa3236ce25dd3347dab8

Creation date: 2022-10-07T17:37:38.974Z

Asignee: user_id

saber

**Fig. 3.** Requests page view in chatbot retraining responsive web app (2022).

## 4 Front-End Implementation

For the implementation of the front-end in conjunction with NDS Cognitive Labs, we work mainly with Javier A. Valdivia on the functionalities required in the system.

Under the framework discussed in this article, we seek to make an implementation of the front-end part, the visual part of the program, that is adaptable to different screen resolutions. See Fig. 2. and 3. And from which you can access different editable options of the chatbot, and even change chatbots to be retrained.

The design proposal is created by NDS Cognitive Labs, and we are entrusted with the technical functionalities of the visual elements, such as buttons, menus, browsers and user interaction elements.

As functional components are built, especially those that are repeated throughout the application, individual pages are then built, which use some of these same components and others to execute their required functionalities.

## 5 Conclusions

In summary, we underline the importance of the chatbot retraining phase in the life cycle of these computer programs, and in this way the focus on the program developed for this same function is justified.

Although many different technologies have been selected and are even used in part for the construction of this system, ReactJS takes a leading role due to its ease of implementation and ability to interact with the Back-End implemented with MongoDB.

Together with NDS Cognitive Labs, specially led by Kevin Zazueta and Ricardo Cantú, the implementation team has been able to bring to reality an interface capable of semi-automating the chatbot retraining process. In addition to including parameterized functionalities to be able to adapt to different systems as needed.

## References

1. Artyco: Chatbots: the customer service revolution. https://artyco.com/los-chatbots-la-revolucion-de-la-atencion-al-cliente (2022)
2. Brush, K., Scardina, J.: Chatbot. TechTarget. https://www.techtarget.com/searchcustomer experience/definition/chatbot (2022)
3. Bautista, I.:Backend and Frontend, What is it and how do they work in programming?https://www.servnet.mx/blog/backend-y-frontend-partes-fundamentales-de-la-programaci on-de-una-aplicacion-web (2022)
4. Enzyme: Chatbot retraining: essential to help the user. Available: https://enzyme.biz/blog/reentrenode-chatbots-imprescindible-para-ayudar-al-usuario (2022)
5. GFCGlobal: What are web applications? https://edu.gcfglobal.org/es/informatica-basica/que-son-las-aplicaciones-web/1/ (2022)

ISSN 1870-4069

# Suicide Tendency Content Detection with Natural Language Processing and LIME Explainer

Alberto Alvarado Sandoval[1], Fernando Aguilar-Canto[2],
Diana Jiménez[2], Hiram Calvo[2]

[1] Instituto Politécnico Nacional,
Escuela Superior de Cómputo,
Mexico

[2] Instituto Politécnico Nacional,
Centro de Investigación en Computación,
Mexico

{alberto.alvarado.sandoval, pherjev,
dianaljl.99, hiramcalvo}@gmail.com

**Abstract.** Suicide is a widespread global concern, particularly among the young population with high social media usage. Detecting suicide tendencies early can offer crucial aid. This study investigates the use of Natural Language Processing (NLP) techniques, including Language Models, to identify suicide-related content. By training a language model on a dataset containing both public content and social media posts from individuals with documented suicide tendencies, we aim to develop a tool for recognizing language patterns suggestive of potential suicide risk. Additionally, we explore the integration of the LIME explainer to enhance local interpretability, improving model comprehension. This paper provides a comprehensive exploration of identifying suicide-related text using NLP, employing diverse methodologies including classical machine learning and state-of-the-art Large Language Models (LLMs) like BERT, RoBERTa, and DistilBERT. Remarkably, the DistilBERT model surpasses more complex counterparts, acchieving 0.97741 of validation accuracy and 0.97584 of testing accuracy. Introducing an explainer algorithm improves model transparency, illuminating decision processes. Our findings emphasize the potential of advanced NLP techniques for understanding and addressing suicide-related content, benefiting mental health professionals and digital platforms striving to offer timely support and intervention.

**Keywords:** Suicide content detection in texts, large language models, logistic regression, random forest

## 1 Introduction

Suicide is a concerning global public health issue. According to the World Health Organization (WHO), approximately 700,000 people take their own lives every year, making it a leading cause of death, especially among young populations aged 15 to 29 years [23]. One crucial aspect of suicide prevention is recognizing the warning signs displayed by individuals at risk [25, 8].

These signs often manifest in their language and expressions, and with the widespread use of the Internet and social media, such content has become more accessible for analysis [6]. In recent years, Natural Language Processing (NLP) techniques, particularly Language Models, have shown remarkable capabilities in understanding and generating human-like text [3]. These Language Models have been applied to a wide range of tasks, from language translation to sentiment analysis.

Recognizing the potential for early detection of suicide tendencies in online content, we propose a research project that leverages Language Models to identify and analyze linguistic markers associated with suicide risk. The objective of this study is to investigate the viability of employing Language Models and NLP techniques for detecting suicide tendencies. By training a language model on a dataset comprising public content and social media posts from individuals with documented suicide tendencies, we aim to create a tool that effectively identifies language patterns indicative of potential suicide risk.

This tool could be integrated into social media platforms and online communities to automatically flag concerning content and offer timely support and resources to those in need. Additionally, we are considering integrating the LIME explainer to provide local explainability, aligning with one of the key directions indicated by Ji et al. (2020) [15]. In this paper, we showcase a range of machine learning models, including language models, for classifying text as either containing suicide-related content or not.

Despite the prevalence of deep learning in this domain, the utilization of language models with explainability remains limited, which is our main contribution to this topic. By highlighting the potential of machine learning models in detecting suicide-related content, our aim is to contribute to suicide prevention initiatives and foster a more supportive online environment for vulnerable individuals.

## 2 Related work

The detection of suicide-related content in texts is a challenging task in Natural Language Processing (NLP). Various techniques, including simple keyword detection [12, 30, 13] and sophisticated Deep Learning classifiers, have been explored by different authors. Suicide-related keywords, such as “kill,” “suicide,” and “depressed,” are associated with intense negative emotions like anxiety and hopelessness, as well as social factors like family and friends [15].

Ji et al. (2018) [16] investigated several algorithms, including Support Vector Machines, dense neural networks, random forest, XGBoost, and Long-Short Term Memory (LSTM) models, using data from Reddit SuicideWatch and Twitter. The XGBoost classifier outperformed other approaches, achieving an F1-score of 0.9583 and an AUC of 0.9569.

Other authors have also explored recurrent networks, such as Coppersmith [6], who used a bidirectional LSTM with self-attention and GloVe word embeddings. Shing et al. [28] used data from Reddit SuicideWatch to label posts according to perceived risk and implemented Convolutional Neural Networks (ConvNets) for classification. Gaur et al. [9] achieved improved results by enriching ConvNets with knowledge bases in measuring suicide risk tendency.

Tadesse et al. [29] compared Deep Learning models (LSTM and ConvNets) with classical approaches, obtaining better results with the Deep Learning techniques. Mohammadi et al. [21] combined ConvNets with different recurrent networks (LSTMs, Bidirectional LSTMs, GRU) and a Support Vector Machine meta-classifier for suicide risk assessment, excelling in subtasks A and C.

Matero et al. [20] presented a suicide risk assessment approach using Bidirectional Transformers for Language Understanding (BERT) embeddings with a neural dual-context model based on two GRU networks with attention, achieving better results in subtask B. Kodati and Ramakrishnudu [17] also used BERT embeddings with recurrent networks (LSTMs, GRUs) and ConvNets, reaching a maximum F1-score of 0.959 in their Reddit dataset.

Benton, Mitchell, and Hovy [2] employed multitask learning (MTL) to detect mental health issues, including suicide attempts, using Twitter data. Ophir et al. [22] developed Single Task and Multiple Task Models to predict suicide risk from Facebook posts, achieving better results with the Multiple Task Model. Both models utilized ELMo embeddings for word vectorization. Ji et al. (2020) [15] implemented Relation Networks with Attention in the UMD Reddit Suicidality Dataset, outperforming other Deep Learning models (ConvNets, LSTMs, Bidirectional LSTMs) in predicting risk levels with three labels indicating the level of risk.

### 2.1 Transformers

With the recent success of the Transformer architecture in various NLP tasks, authors have applied it to the suicide classification problem. Haque et al. [10] fine-tuned Language Models (BERT, ALBERT, RoBERTa, XLNET) and achieved better results than a Bidirectional LSTM. RoBERTa attained an F1-score of 0.9547 on Reddit data. Similarly, Ananthakrishnan et al. [1] implemented BERT, DistilBERT, ALBERT, RoBERTa, and DistilRoBERTa on a Twitter dataset, with RoBERTa yielding the best results.

Additionally, Zhang et al. (2021) [31] developed the TransformerRNN architecture, outperforming other classifiers, including Naive Bayes, ConvNets, and various LSTMs, including a Bidirectional LSTM with attention, in classifying suicide, last statements from executed prisoners, and neutral posts. Sawhney et al. [27] introduced STATENet, a time-aware transformer-based model, for identifying suicidal intent in English tweets by incorporating historical emotional context.

The model outperforms other methods (ConvNets, random forest, a BERT-based classifier), highlighting the importance of emotional and temporal cues in assessing suicide risk on social media. Burkhardt [4] evaluates the utility of social media-derived training data for suicide risk prediction in clinical settings and develops a metric for assessing the clinical usefulness of automated triage.

Their BERT-based model with multi-stage transfer learning improves suicide risk prediction, achieving a F1-score of 0.797 in a Reddit dataset. The study demonstrates the potential of leveraging social media data for better risk assessment and improved clinical outcomes in suicide prevention interventions. For languages other than English, Hassib et al. [11] focus on identifying depression and suicidal ideation in the Middle East, specifically in Egypt, where suicidal deaths are prevalent.

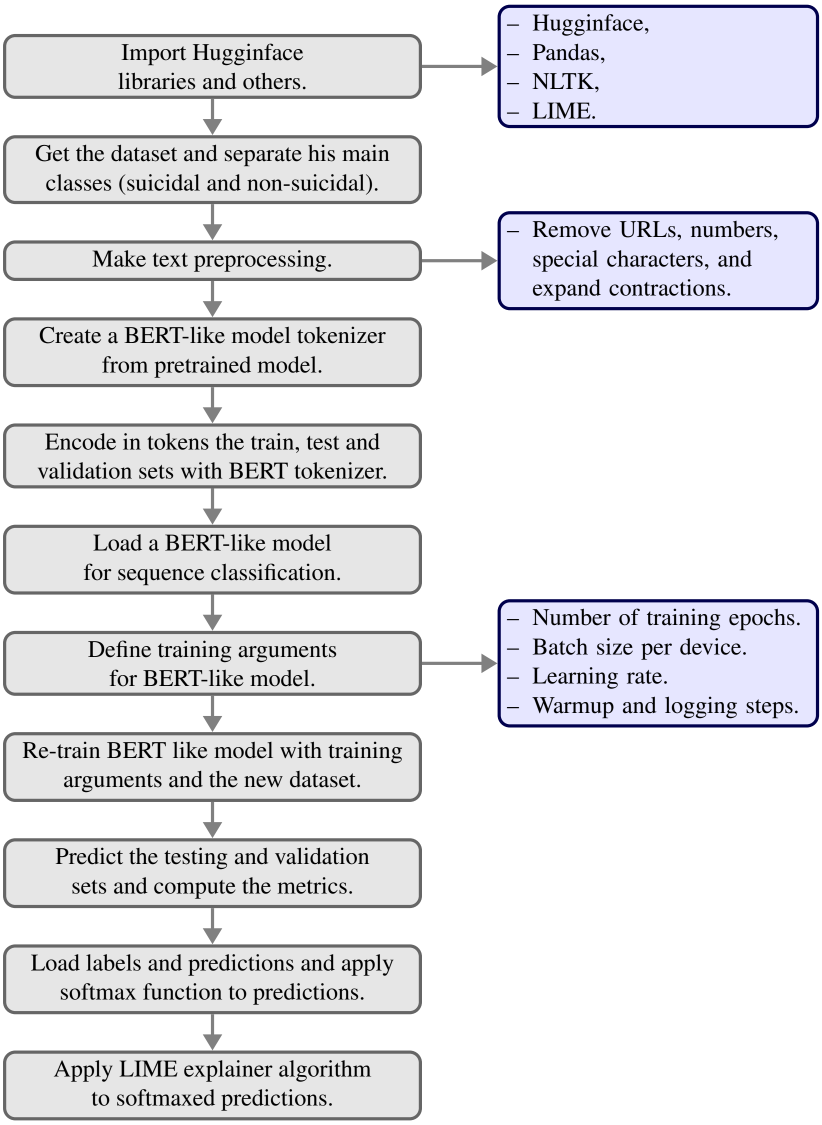

Import Hugginface libraries and others.
– Hugginface,
– Pandas,
– NLTK,
– LIME.
Get the dataset and separate his main classes (suicidal and non-suicidal).
Make text preprocessing.
– Remove URLs, numbers, special characters, and expand contractions.
Create a BERT-like model tokenizer from pretrained model.
Encode in tokens the train, test and validation sets with BERT tokenizer.
Load a BERT-like model for sequence classification.
Define training arguments for BERT-like model.
– Number of training epochs.
– Batch size per device.
– Learning rate.
– Warmup and logging steps.
Re-train BERT like model with training arguments and the new dataset.
Predict the testing and validation sets and compute the metrics.
Load labels and predictions and apply softmax function to predictions.
Apply LIME explainer algorithm to softmaxed predictions.

**Fig. 1.** Diagram of the process.

Due to a lack of mental health awareness in Arabic culture, the authors utilize social media, particularly Twitter, where users express emotions openly. They create the AraDepSu dataset with three classes (“depressed,” “suicidal,” and “neutral”) from manually labeled tweets and train it on various Transformer-based models. MARBERT performs the best, achieving high accuracy and macro-average F1-score values of 0.9120 and 0.8875, respectively. It is important to note that this study targets a different dataset and topic than the Reddit SuicideWatch revision, making direct comparisons challenging, and privacy preservation for users is a significant concern.

**Table 1.** Main results.

| Model | Validation Accuracy | Testing Accuracy |
|---|---|---|
| Logistic Regression | 0.92493 | 0.92691 |
| Random Forest | 0.90776 | 0.90801 |
| BERT-base-uncased | 0.97641 | 0.97398 |
| RoBERTa-base | 0.97544 | 0.81123 |
| DistilBERT-base-uncased | **0.97751** | **0.97584** |

# 3 Methodology

We utilized the Hugging Face transformers library for certain parts of text preprocessing and model development. All the methodology is summarized in figure 1. The text preprocessing involved several steps, including converting all text to lowercase, removing HTML tags, URLs, numbers, punctuation, and single characters. Contractions were replaced with their expanded forms, and all stopwords were removed. The texts were tokenized using three model tokenizers:

1. BERT [7],
2. DistilBERT [26],
3. RoBERTa [19].

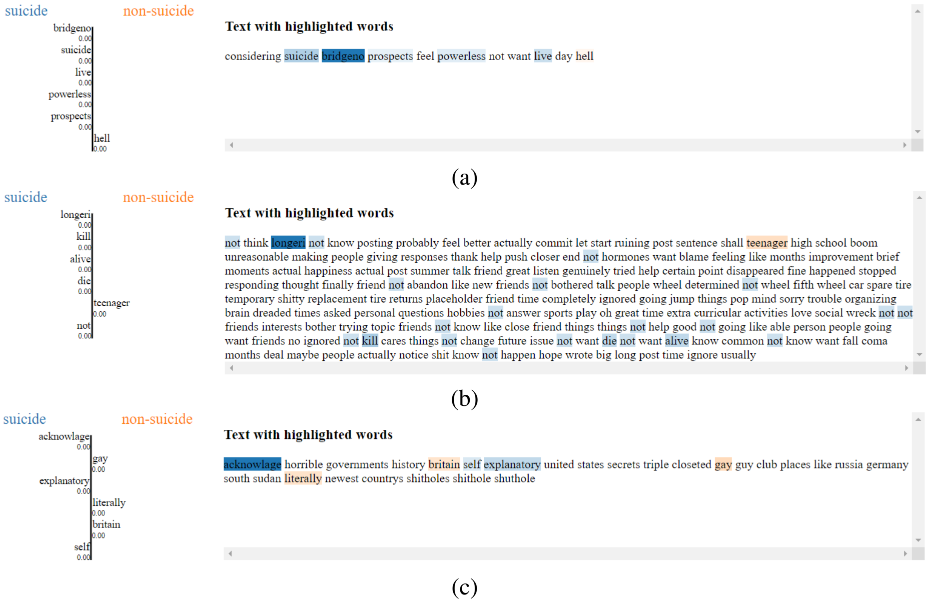
The models were fine-tuned with one training epoch, a learning rate of 0.00005, 10 warm-up steps, and the F1-score as the training metric. Test results were predicted using each model, and the outcomes are presented below. In addition, we incorporated the Logistic Regression and Random Forest algorithms (with 100 estimators) from the Scikit-learn library to conduct a comparative analysis of its performance against that of the Large Language Models. To facilitate this evaluation, we employed the Tf-Idf Vectorizer as the feature extraction method.

## 3.1 Dataset description

The dataset used in this study was obtained from the “SuicideWatch” and “Depression” subreddits on Reddit, and it is available on [18]. It consists of 232,074 posts collected using the Pushift API from December 16, 2008, to January 2, 2021. The dataset is well-suited for the research because of its large size, balanced class distribution, and organic nature of the text from Reddit posts. It contains 116,037 labeled as suicidal and 116,037 as non-suicidal texts, which is one the largest NLP datasets of the topic (see [14]). 2/3 of the dataset were devoted to the training set, whereas 2/9 were used for validation and 1/9 for testing.

# 4 Results

The table 1 presents the main results of the study. Various models were evaluated using both validation and testing datasets. The logistic regression model achieved a high accuracy of 92.49% on the validation set and 92.69% on the testing set.



**Text with highlighted words**

considering suicide bridgeno prospects feel powerless not want live day hell

(a)

**Text with highlighted words**

not think longeri not know posting probably feel better actually commit let start ruining post sentence shall teenager high school boom unreasonable making people giving responses thank help push closer end not hormones want blame feeling like months improvement brief moments actual happiness actual post summer talk friend great listen genuinely tried help certain point disappeared fine happened stopped responding thought finally friend not abandon like new friends not bothered talk people wheel determined not wheel fifth wheel car spare tire temporary shitty replacement tire returns placeholder friend time completely ignored going jump things pop mind sorry trouble organizing brain dreaded times asked personal questions hobbies not answer sports play oh great time extra curricular activities love social wreck not not friends interests bother trying topic friends not know like close friend things things not help good not going like able person people going want friends no ignored not kill cares things not change future issue not want die not want alive know common not know want fall coma months deal maybe people actually notice shit know not happen hope wrote big long post time ignore usually

(b)

**Text with highlighted words**

acknowlage horrible governments history britain self explanatory united states secrets triple closeted gay guy club places like russia germany south sudan literally newest countrys shitholes shithole shuthole

(c)

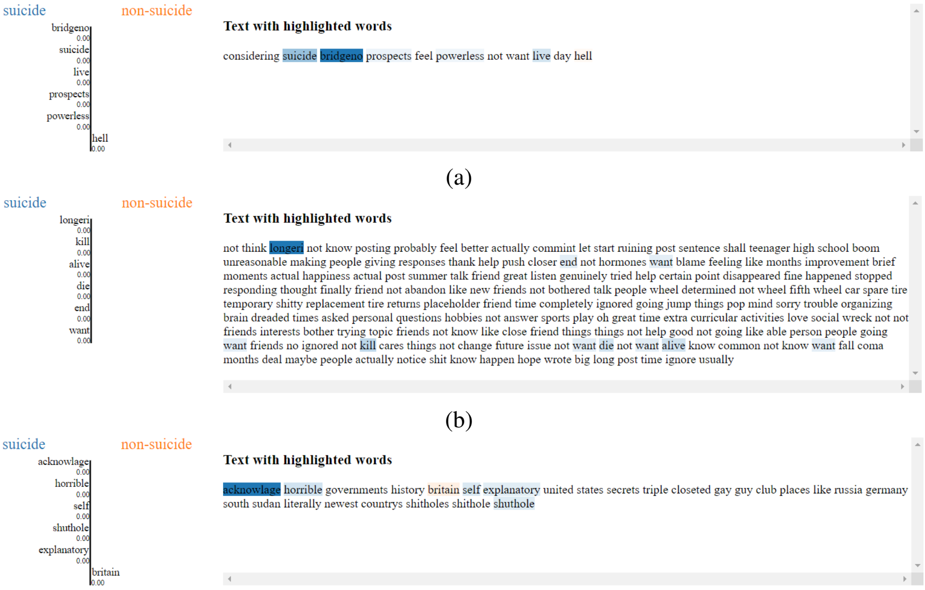
**Fig. 2.** Examples of the outputs of LIME with the best model (DistilBERT) with samples of the testing set.

The random forest model demonstrated slightly lower accuracy with 90.77% on validation and 90.80% on testing. Notably, BERT-base-uncased exhibited remarkable performance, achieving 97.64% accuracy on validation and 97.40% accuracy on testing. In contrast, RoBERTa-base displayed a high validation accuracy of 97.54%, but its testing accuracy dropped to 81.12%. The DistilBERT-base-uncased model outperformed the rest, boasting the highest validation accuracy of 97.75% and testing accuracy of 97.58%.

## 4.1 Explainability

To increase model transparency, we integrated the Local Interpretable Model-agnostic Explanations (LIME) explainer [24], which faithfully interprets predictions by locally approximating the model with an interpretable one. We gathered 6000 samples for the explainer algorithm. Figure 2 depicts the application of LIME to the optimal model, DistilBERT, while Figure 3 showcases the results of the explainer applied to the second-best model, BERT.

In the initial sentence (see Figure 2a and 4.1), the DistilBERT classifier accurately categorizes the text as "suicide." The words "suicide," "bridgeno," "prospects," "powerless," and "live" are unsurprisingly associated with the "suicide" class. However, unexpectedly, the word ”hell” is linked to the negative category. The same results were observed in BERT. Moving on to the subsequent sentence (Figures 2b and 4.1), in the case of DistilBERT, words such as "not," "longer," "kill," "alive," and "die" exhibit a strong association with the "suicide" class, while "teenager" is associated with the non-suicidal category.



**Text with highlighted words**

considering suicide bridgeno prospects feel powerless not want live day hell

(a)

**Text with highlighted words**

not think longeri not know posting probably feel better actually commint let start ruining post sentence shall teenager high school boom unreasonable making people giving responses thank help push closer end not hormones want blame feeling like months improvement brief moments actual happiness actual post summer talk friend great listen genuinely tried help certain point disappeared fine happened stopped responding thought finally friend not abandon like new friends not bothered talk people wheel determined not wheel fifth wheel car spare tire temporary shitty replacement tire returns placeholder friend time completely ignored going jump things pop mind sorry trouble organizing brain dreaded times asked personal questions hobbies not answer sports play oh great time extra curricular activities love social wreck not not friends interests bother trying topic friends not know like close friend things things not help good not going like able person people going want friends no ignored not kill cares things not change future issue not want die not want alive know common not know want fall coma months deal maybe people actually notice shit know happen hope wrote big long post time ignore usually

(b)

**Text with highlighted words**

acknowlage horrible governments history britain self explanatory united states secrets triple closeted gay guy club places like russia germany south sudan literally newest countrys shitholes shithole shuthole

(c)

**Fig. 3.** Examples of the outputs of LIME with the second best model (BERT) with samples of the testing set.

A similar pattern emerges with BERT, although it does not attribute negative associations to the "suicide" class. Instead, it considers the words "end" and "wants" as associated with this class. Both models correctly classify this text as "suicidal." Finally, applying DistilBERT in the last example reveals that words like "gay," "literally," and "britain" correlate with the non-suicidal class, whereas "acknowlage," "explanatory," and "self" are linked to the "suicide" class.

Consequently, this example is correctly classified as "non-suicidal" by DistilBERT. On the other hand, the BERT model identifies "acknowlage" and "explanatory" as linked to the suicide class but adds the words "shuthole" and "horrible" to this category. It only considers the word "britain" as associated with the non-suicidal class. While this overview is not exhaustive, it provides insights into the pertinent words considered by each model, aligning with the observations made by Ji et al. (2020) [15].

## 5 Discussion

This study addresses the pressing need to employ modern language models in addressing the issue, echoing the call made by Ji et al. [15]. While explainer algorithms have found application in the context of depression detection [5], their utility in the suicide detection problem remains relatively unexplored. Leveraging explainers like LIME can significantly benefit professionals seeking to comprehend the decision-making processes within Deep Learning models.

Turning to the numerical results, it is important to acknowledge that while many related works have utilized data from Reddit, direct comparisons are challenging due to variations in datasets. However, as discussed in Section 2, even the most promising numerical outcomes have yielded F1-scores below 0.96. This performance metric falls short of the results achieved by DistilBERT in our study.

This suggests that the adoption of advanced language models, especially DistilBERT, holds significant promise in enhancing suicide detection capabilities, potentially offering improved support and intervention for at-risk individuals. Nonetheless, the need for continued research and exploration in this vital area is evident, particularly in expanding the utility of explainer algorithms and fine-tuning models for real-world applicability.

## 6 Conclusions

In this study, we have tackled the task of detecting text related to suicide tendencies, framing it as a natural language processing (NLP) classification problem. We explored the effectiveness of various machine learning approaches, including classical algorithms like Logistic Regression and Random Forest, alongside state-of-the-art Large Language Models (LLMs) such as BERT, RoBERTa, and DistilBERT. As anticipated, our experiments revealed that the deep learning-based LLMs generally outperformed the classical methods, underscoring the power of these advanced models in capturing intricate language patterns indicative of suicide-related content.

However, the unexpected outcome emerged where DistilBERT, a distilled version of BERT, exhibited superior performance compared to BERT and even RoBERTa, a larger model. This observation highlights the intricate interplay between model complexity and performance, indicating that more extensive architectures might not always lead to better results.

Building upon the insights provided by [15], we incorporated an explainer algorithm to demystify the decision-making process of our best-performing model, DistilBERT. By doing so, we aimed to enhance the interpretability of the algorithm, which could be crucial for professionals and practitioners dealing with this critical problem. The fusion of fine-tuned LLMs with explainer algorithms holds promise for the future, offering a valuable tool to understand, analyze, and address suicide tendency content more effectively.

## References

1. Ananthakrishnan, G., Kumar-Jayaraman, A., Trueman, T. E., Mitra, S., Abinesh, A. K., Murugappan, A.: Suicidal intention detection in tweets using BERT-based transformers. In: International Conference on Computing, Communication, and Intelligent Systems, pp. 322–327 (2022) doi: 10.1109/icccis56430.2022.10037677
2. Benton, A., Mitchell, M., Hovy, D.: Multitask learning for mental health conditions with limited social media data. In: Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, vol. 1, pp. 152–162 (2017)

3. Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D., Wu, J., Winter, C., Hesse, C., et al.: Language models are few-shot learners. Advances in Neural Information Processing Systems 33, vol. 33, pp. 1877–1901 (2020)
4. Burkhardt, H. A., Ding, X., Kerbrat, A., Comtois, K. A., Cohen, T.: From benchmark to bedside: Transfer learning from social media to patient-provider text messages for suicide risk prediction. Journal of the American Medical Informatics Association, vol. 30, no. 6, pp. 1068–1078 (2023) doi: 10.1093/jamia/ocad062
5. Byeon, H.: Advances in machine learning and explainable artificial intelligence for depression prediction. International Journal of Advanced Computer Science and Applications, vol. 14, no. 6 (2023) doi: 10.14569/ijacsa.2023.0140656
6. Coppersmith, G., Leary, R., Crutchley, P., Fine, A.: Natural language processing of social media as screening for suicide risk. Biomedical Informatics Insights, vol. 10 (2018) doi: 10.1177/1178222618792860
7. Devlin, J., Chang, M. W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, vol. 1, pp. 4171–4186 (2018)
8. Dilillo, D., Mauri, S., Mantegazza, C., Fabiano, V., Mameli, C., Zuccotti, G. V.: Suicide in pediatrics: Epidemiology, risk factors, warning signs and the role of the pediatrician in detecting them. Italian Journal of Pediatrics, vol. 41, no. 1 (2015) doi: 10.1186/s13052-015 -0153-3
9. Gaur, M., Alambo, A., Sain, J. P., Kursuncu, U., Thirunarayan, K., Kavuluru, R., Sheth, A., Welton, R., Pathak, J.: Knowledge-aware assessment of severity of suicide risk for early intervention. In: Proceedings of the World Wide Web Conference, pp. 514–525 (2019) doi: 10.1145/3308558.3313698
10. Haque, F., Nur, R. U., Al Jahan, S., Mahmud, Z., Shah, F. M.: A transformer based approach to detect suicidal ideation using pre-trained language models. In: Proceedings of the 23rd International Conference on Computer and Information Technology, pp. 1–5 (2020) doi: 10.1109/iccit51783.2020.9392692
11. Hassib, M., Hossam, N., Sameh, J., Torki, M.: AraDepSu: Detecting depression and suicidal ideation in arabic tweets using transformers. In: Proceedings of the 7th Arabic Natural Language Processing Workshop, pp. 302–311 (2022) doi: 10.18653/v1/2022.wanlp-1.28
12. Huang, Y. P., Goh, T., Liew, C. L.: Hunting suicide notes in web 2.0-preliminary findings. In: Proceedings of the 9th IEEE International Symposium on Multimedia Workshops, pp. 517–521 (2007) doi: 10.1109/ism.workshops.2007.92
13. Jashinsky, J., Burton, S. H., Hanson, C. L., West, J., Giraud-Carrier, C., Barnes, M. D., Argyle, T.: Tracking suicide risk factors through twitter in the US. Crisis, vol. 35, no. 1, pp. 51–59 (2014) doi: 10.1027/0227-5910/a000234
14. Ji, S., Li, X., Huang, Z., Cambria, E.: Suicidal ideation and mental disorder detection with attentive relation networks. Neural Computing and Applications, vol. 34, no. 13, pp. 10309–10319 (2021) doi: 10.1007/s00521-021-06208-y
15. Ji, S., Pan, S., Li, X., Cambria, E., Long, G., Huang, Z.: Suicidal ideation detection: A review of machine learning methods and applications. IEEE Transactions on Computational Social Systems, vol. 8, no. 1, pp. 214–226 (2021) doi: 10.1109/tcss.2020.3021467
16. Ji, S., Yu, C. P., Fung, S. F., Pan, S., Long, G.: Supervised learning for suicidal ideation detection in online user content. Complexity, vol. 2018, pp. 1–10 (2018) doi: 10.1155/2018 /6157249

17. Kodati, D., Tene, R.: Identifying suicidal emotions on social media through transformer-based deep learning. Applied Intelligence, vol. 53, no. 10, pp. 11885–11917 (2023) doi: 10.1007/s10489-022-04060-8
18. Komati, N.: Kaggle: Competitors contributor (2021) www.kaggle.com/nikhileswarkomati
19. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V.: RoBERTa: A robustly optimized BERT pretraining approach. arXiv, (2019) doi: 10.48550/arXiv.1907.11692
20. Matero, M., Idnani, A., Son, Y., Giorgi, S., Vu, H., Zamani, M., Limbachiya, P., Guntuku, S. C., Schwartz, H. A.: Suicide risk assessment with multi-level dual-context language and BERT. In: Proceedings of the 6th Workshop on Computational Linguistics and Clinical Psychology, pp. 39–44 (2019) doi: 10.18653/v1/w19-3005
21. Mohammadi, E., Amini, H., Kosseim, L.: CLaC at CLPsych 2019: Fusion of neural features and predicted class probabilities for suicide risk assessment based on online posts. In: Proceedings of the 6th Workshop on Computational Linguistics and Clinical Psychology, pp. 34–38 (2019) doi: 10.18653/v1/W19-3004
22. Ophir, Y., Tikochinski, R., Asterhan, C. S. C., Sisso, I., Reichart, R.: Deep neural networks detect suicide risk from textual facebook posts. Scientific Reports, vol. 10, no. 1 (2020) doi: 10.1038/s41598-020-73917-0
23. Organization, W. H.: Suicide (2021) www.who.int/news-room/fact-sheets/detail/suicide
24. Ribeiro, M. T., Singh, S., Guestrin, C.: Why should i trust you? explaining the predictions of any classifier. In: Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations, pp. 97–101 (2016) doi: 10.18653/v1/N16-3020
25. Rudd, M. D.: Suicide warning signs in clinical practice. Current Psychiatry Reports, vol. 10, no. 1, pp. 87–90 (2008) doi: 10.1007/s11920-008-0015-4
26. Sanh, V., Debut, L., Chaumond, J., Wolf, T.: Distilbert, a distilled version of BERT: Smaller, faster, cheaper and lighter. arXiv, (2019) doi: 10.48550/ARXIV.1910.01108
27. Sawhney, R., Joshi, H., Gandhi, S., Shah, R.: A time-aware transformer based model for suicide ideation detection on social media. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, pp. 7685–7697 (2020) doi: 10.18653/v1/2020.emnlp-main.619
28. Shing, H. C., Nair, S., Zirikly, A., Friedenberg, M., Daumé, H., Resnik, P.: Expert, crowdsourced, and machine assessment of suicide risk via online postings. In: Proceedings of the 5th Workshop on Computational Linguistics and Clinical Psychology: From Keyboard to Clinic, pp. 25–36 (2018) doi: 10.18653/v1/W18-0603
29. Tadesse, M. M., Lin, H., Xu, B., Yang, L.: Detection of suicide ideation in social media forums using deep learning. Algorithms, vol. 13, no. 1, pp. 7 (2019) doi: 10.3390/a13010007
30. Varathan, K. D., Talib, N.: Suicide detection system based on twitter. In: Proceedings of the Science and Information Conference, pp. 785–788 (2014) doi: 10.1109/sai.2014.6918275
31. Zhang, T., Schoene, A. M., Ananiadou, S.: Automatic identification of suicide notes with a transformer-based deep learning model. Internet Interventions, vol. 25, pp. 100422 (2021) doi: 10.1016/j.invent.2021.100422

ISSN 1870-4069

# Bayesian Learning for User Modeling

Rebai Rim[1], Maalej Mohamed Amin[1,2],
Adel Mahfoudhi[1,2]

University of Sfax,
Ecole Nationale d'Ingénieurs de Sfax,
Tunisia

University of Sfax,
Faculté des Sciences de Sfax,
Tunisia

corre@prueba.com
correo@prueba.com

**Abstract.** User modeling has been a focal point of numerous studies in the web domain. The World Wide Web serves as a widely utilized platform for delivering adaptive user interfaces. Various techniques have been explored to capture user preferences, with Bayesian Networks (BNs) emerging as an effective means of constructing probabilistic models. This paper introduces an adaptive user interface, specifically designed for web applications, focusing on a social network context. The construction of our Bayesian user model is detailed, involving a comparison of learning algorithms to train the Bayesian structure. Evidence in a Bayesian network serves as a pivotal point for inference methods, originating from information derived through variable observations. Subsequently, inference algorithms are employed to enable the user model to predict user preferences. The effectiveness of the Bayesian Network in predicting user preferences is affirmed by evaluating inferred results for essential variables across various scenarios. Ultimately, the adaptive user interface is validated as more user-friendly compared to a fixed user interface.

**Keywords:** Learning model, user modeling, Bayesian model.

## 1 Introduction

Numerous research endeavors are dedicated to exploring the ergonomics and information organization within an interface, along with adapting this information to align with user preferences. Central to issues of presentation and adaptive interfaces are the user's preferences and experience. Users possess diverse interests, knowledge, learning styles, and preferences, leading to a significant focus on designing interfaces capable of understanding user characteristics.

A crucial area of research involves determining how interfaces can be crafted to comprehend user attributes. To deliver personalized information, it becomes imperative to observe and generalize from user behavior, enabling predictions based on these observations. This collection of information about the user is commonly referred to as a user model [15, 33].

The objective of a user model may encompass predicting user behavior to acquire user knowledge or creating a comprehensive database of user profiles. To address these requirements, contemporary research on user modeling is shifting towards more comprehensive models that incorporate uncertain knowledge. Traditional models are deemed inadequate in capturing the intricacies of the human-machine interface.

Consequently, Bayesian Networks are gaining prominence as a means to handle uncertainty in user modeling [14, 40]. This trend is attributed to their lucid semantics and the opportunities they present for machine learning [42, 16]. In this study, the Bayesian networks approach was implemented within a web interface to infer user preferences. Our approach leveraged learning and inference algorithms, with the obtained results showcasing the effectiveness of Bayesian networks in user modeling.

Evidence forms the foundation for inference methods. In certain studies [39, 1, 6, 5, 13, 23, 28, 55], the term “soft evidence” is employed to denote probabilistic evidence [21]. However, in another study [7, 11], the use of the term “soft evidence” was abandoned. This literature review, coupled with an examination of Bayesian network software, reveals the inconsistent usage of the term “soft evidence,” leading to confusion.

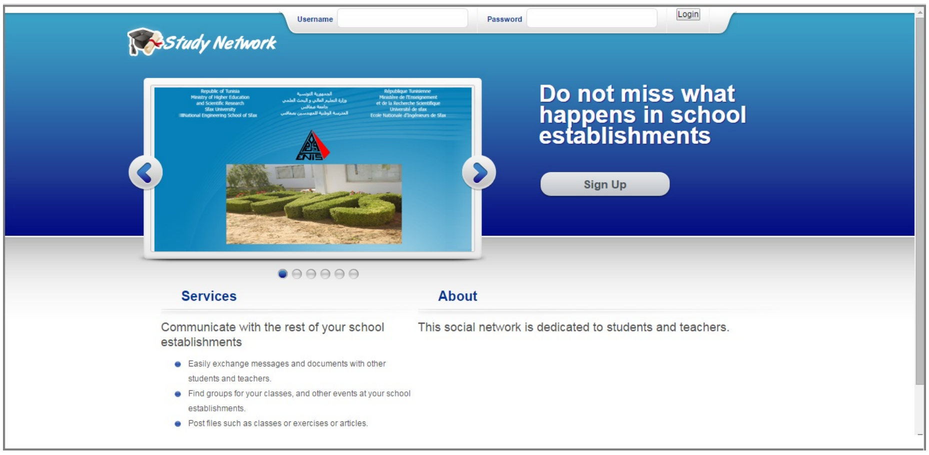
Notably, Valtorta and co-authors [22, 41, 30, 29, 44, 43, 52, 37, 38, 4], along with [27, 51], refer to probabilistic evidence as “soft evidence”. The proposed methodology underwent evaluation, involving ten subjects performing tasks using both a fixed user interface and the proposed adaptive user interface. The remainder of this paper is structured as follows: Section 2 introduces the related work.

Bayesian Rules was defined in Section 2. Section 3 is dedicated to defining the construction of the Bayesian networks model, utilizing learning algorithms. Section 4 and 5 focuses on the update of a probabilistic evidence distribution within our Bayesian model. A comprehensive discussion of the experimental results is presented in Section 6 covering the prediction of user needs of our users interface. Finally, Section 7 concludes the paper.

## 2 Bayesian Networks for User Modeling

The exploration of user modeling has been a focal point in numerous studies, reflecting its intricate nature that spans various domains of information and human sciences, including psychology, education, artificial intelligence, and human-machine interface. This multifaceted field has yielded a mix of successes and failures. The surge in interest in the 1980s, particularly within the Intelligent Tutoring Systems (ITS) domain, marked a significant growth in user modeling research [3, 45].

Advancements in existing user models have aimed to enhance personalization capabilities by considering diverse user characteristics. A substantial body of research has concentrated on delivering personalized services across a range of web applications, such as e-commerce [53], books[36], films [17] and online research papers [34]. Notably, Jie and al [31] emphasized the integration of a hybrid fuzzy semantic recommendation approach into an intelligent recommendation system, enabling the personalized recommendation of relevant business partners to individual users.


Username
Password
Login
Study Network
Republic of Tunisia
Ministry of Higher Education
and Scientific Research
Sfax University
National Engineering School of Sfax
République Tunisienne
Ministère de l'Enseignement
et de la Recherche Scientifique
Université de sfax
Ecole Nationale d'Ingénieurs de Sfax
ENIS
Sign Up

Do not miss what happens in school establishments

Services

Communicate with the rest of your school establishments

- Easily exchange messages and documents with other students and teachers.
- Find groups for your classes, and other events at your school establishments.
- Post files such as classes or exercises or articles.

About

This social network is dedicated to students and teachers.

**Fig. 1.** The web interface.

Probabilistic models have been instrumental in addressing the uncertainty associated with modeling user needs and characteristics, making them well-suited for the challenges inherent in user modeling [52, 2]. In particular, Bayesian user modeling has emerged as an effective method for understanding a user's objectives [3, 9, 10]. They have been applied in various systems to anticipate user preferences and objectives [19, 48, 50]. Horvitz and al [19] employed Bayesian networks to deduce users' objectives and needs based on their interactions and conditional probability models.

In Horvitz's study, a Bayesian model was initially constructed by psychologists through observations of users in different situations. In another research endeavor [20], Horvitz and al introduced a system designed to forecast a user's intention within an uncertain environment. Various approaches have been explored in the domain of student modeling [3, 8]. Notably, Bayesian networks have garnered significant attention from theorists due to their robust mathematical foundations and their natural representation of uncertainty through probability models [8].

Andes [9], a tutorial employed in e-learning for addressing problems related to Newtonian mechanics, assesses the student's level of knowledge. Millàn and al [35] have equipped education practitioners with the necessary background to comprehend Bayesian networks and applied them to construct student models. Nguyen and Do [40] introduced an approach that combines Bayesian networks and the overlay model, enabling the inference of a user's knowledge from evidence gathered during the learning process. [32] presented various systems employing Bayesian networks to choose the next teaching activity for students based on their knowledge level.

Moreover, Bayesian networks have proven to be a straightforward and effective method for managing uncertainty in context awareness [12]. Korpipaa and al [26] characterized a user's context using the naive Bayes classifier. Hong and al [18] suggested a context-sensitive messenger that autonomously deduces the user's context. In a recent study [49], In-Jee Song and al proposed a model for the ubiquitous family environment to implement a context-adaptive user interface. They employed a Bayesian network to allocate necessary devices in each location.

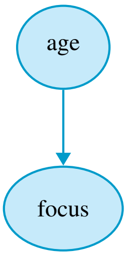


**Fig. 2.** Representation of two nodes.

The knowledge models outlined earlier may not attribute reasons for failures in the knowledge domain. Conversely, preference models solely furnish information about users' preferences without accounting for the evolution of these preferences. This paper introduces an effective approach that tackles the challenging task of detecting user preferences using probabilistic models. Specifically, we developed a Bayesian user model capable of automatically discerning users' preferences.

Our approach considers the evolution of preferences, resulting in a more accurate alignment between a user's preferences and the provided information. Despite the constrained probability of interaction through a web application due to service quality, we successfully gathered user preferences from our adaptive user interface (Figure 1). Subsequently, we created specific models to recommend web interfaces with the most suitable content and format for each user.

## 3 Bayesian Networks

Bayesian networks [25] are graphical models with both qualitative and quantitative components. The qualitative aspect involves the network's structure, represented by a directed acyclic graph where nodes correspond to variables, and arcs depict influences between these variables. On the quantitative side, conditional probability tables define the network settings. These networks serve as potent tools, leveraging algorithms for both inference and associated learning.

Inference relies on Bayes' theorem to disseminate knowledge within the network, while learning encompasses the determination of both the network structure and parameters. These parameters can be derived from either complete or incomplete data. To be more precise, a Bayesian Network constitutes a collection comprising a directed acyclic graph and n random variables $X = X_1, X_2, .., X_n$, establishing a bijection between the set of graph vertices and the set of random variables, ensuring that:

$$P(X_1, X_2, \cdots, X_n) = \prod_{i=1}^{n} P(X_i \mid \mathrm{pa}(X_i)), \tag{1}$$

where, $pa(X_i)$ denotes the collection of parents of variable $X_i$ within the graph.

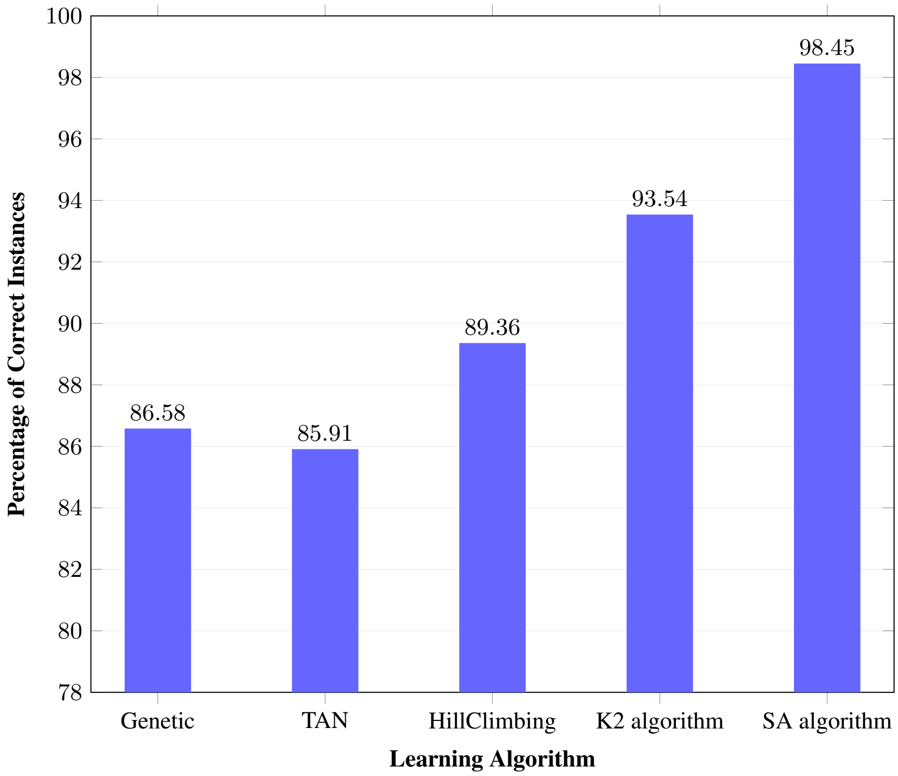


**Fig. 3.** A comparison between the learning algorithms.

## 4 Learning Structure in Bayesian Model

To capture a user's experience in an initial model, we utilized data from our web interface, which functions as a social study network. The user's interactions, stemming from their experiences, were employed to update the parameter set of their model. Depending on the ongoing observations within the Bayesian model, learning and inference algorithms were applied to enhance and refine the Bayesian user model.

Our goal was to identify the pivotal variables in a user model for creating a personalized model for a web interface. To achieve this, instead of treating items for the user as isolated variables, we endeavored to consolidate and represent them in a Bayesian Network model. Implementing rules were then incorporated to articulate these aspects. User preferences, in this context, may encompass: preferences for the presentation of information, selection of services, Determination of which advertisements to display, specification of graphic components to be omitted.

Presuming that our adaptive interface can deduce user preferences, gleaned from observations such as age, interests, gender, and profession, the concept revolves around assigning a profile to each user. We initiate this process by establishing certain nodes of random variables that serve as representations for these profiles.

**Algorithm 1** A pseudocode for the simulated annealing algorithm used to create the structure of a Bayesian model.

```
procedure SIMULATED ANNEALING()
    $D$.                                        ▷ A DataSet.
    $B_{\text{current}}$.                       ▷ Create initial network.
    $T_0$, $T_{\text{final}}$: integer.         ▷ Set the initial temperature.
    $N$.                                        ▷ Make N attempts for the given temperature $T$.
    $t = 1$.
    repeat
        repeat
            Create new network $B_{\text{new}}$ from $B_{\text{current}}$.
            Evaluate $E(B_{\text{current}}, D)$, $E(B_{\text{New}}, D)$.
            if $E(B_{\text{New}}, D) \leq E(B_{\text{current}}, D)$ then
                $B_{\text{current}} = B_{\text{New}}$.
            else
                $X = \frac{E(B_{\text{New}}, D) - E(B_{\text{current}}, D)}{T(t)}$.
                Select a number, denoted as $p$, uniformly at random from the interval [0, 1].
                if $(e^{-x} > p)$ then
                    $B_{\text{current}} = B_{\text{New}}$.
                end if
            end if
            $N = N - 1$.
        until $N = 0$.
        $T_{t+1} = \tau(T_t)$.
        $t = t + 1$.
    until $T_{t+1} \geq T_{\text{final}}$.
    $B_{\text{final}} = B_{\text{current}}$.       ▷ Final Network.
end procedure
```

As an illustration, we depict the node “focus” with a marginal probability vector that signifies the likelihood of the user having a specific center of interest: As an illustration, we depict the node “age” with a marginal probability vector that signifies the age of the user:

$$P(\text{age} = a_1), \cdots, P(\text{age} = a_n) \tag{2}$$

Subsequently, we delineate the array of potential focus $f_1; ...; f_k$. This can be represented by introducing another random variable named “focus”. In this case, we obtain a matrix of conditional probabilities for all possible scenarios:

$$\begin{gathered} P(\text{focus} = f_1|\text{age} = a_1); \cdots; P(\text{focus} = f_1|\text{age} = a_n) \\ \vdots \\ P(\text{focus} = f_k|\text{age} = a_1); \cdots; P(\text{focus} = f_k|\text{age} = a_n) \end{gathered} \tag{3}$$

The decision regarding which focus to present is contingent on the user’s age, prompting us to establish a causal relationship between the two nodes (see Figure 2). Initially, the Bayesian network was constructed based on expert knowledge, even in the absence of training data.

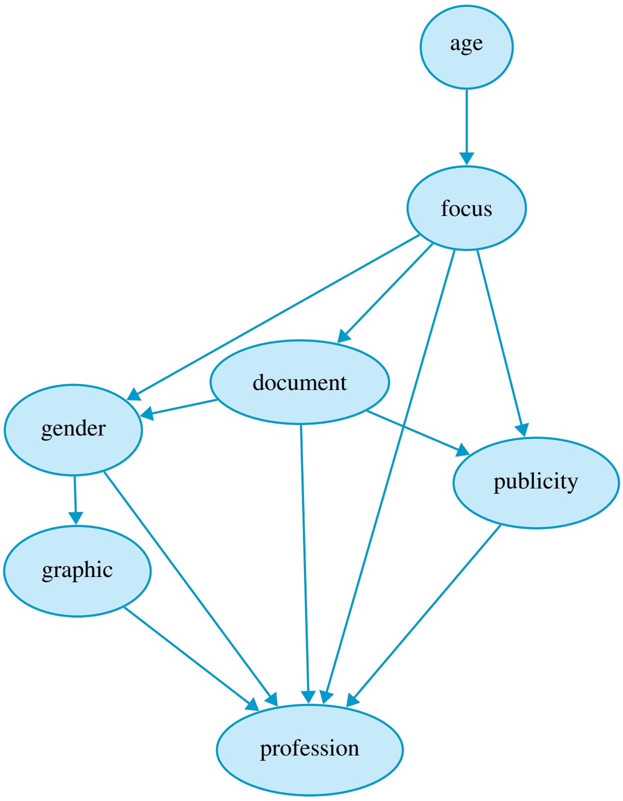


**Fig. 4.** The structure of the Bayesian network.

Once a substantial amount of user data became available, the Bayesian network was trainable. The typical process of training Bayesian networks from data involves two fundamental aspects: training the structure and training the parameter set. The structural training entails searching for the network structure that most closely aligns with the provided data. To identify the optimal structure, learning algorithms leveraging score metrics can be employed. Our database comprised 256 users, indicating its relatively modest size. Consequently, for the testing phase, we opted for the cross-validation process.

This method was employed to assess the predictive efficacy of Bayesian models using a single database. In the cross-validation test, we conducted a comparative analysis of various learning algorithms including simulated annealing algorithm (SA), the K2 algorithm, Tree Augmented Naive Bayes algorithm (TAN), genetic algorithm and Hill Climbing algorithm.

The structure of our Bayesian user model was developed using Weka [54]. The results, detailed in 3, revealed that the predictive performance of the network is contingent on the algorithm employed. Notably, the simulated annealing algorithm exhibited a high accuracy of approximately 98%, while the K2 algorithm performed slightly lower at about 94% correct values.

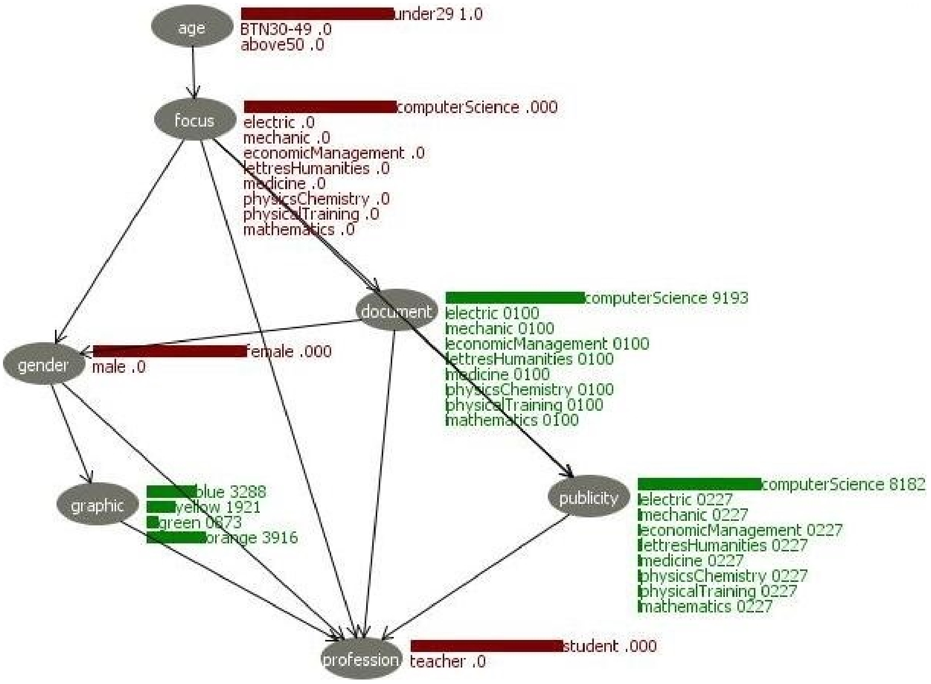


**Fig. 5.** Bayesian model.

The Hill Climbing algorithm yielded around 89% accuracy, and both the genetic and TAN algorithms showed an accuracy of approximately 86%. Therefore, we employed the simulated annealing algorithm to train the structure of our model. Simulated annealing (SA) is a global optimization technique crucial for enabling the current solution to transition to less optimal states based on a probability function.

This feature prevents the algorithm from being confined to a local optimum. Algorithm 1 outlines the simulated annealing algorithm, which draws analogies between the network ( E represented in the structure as the function to be minimized), the configuration of the network structure (B, where E is a function of B), and the temperature (T, an annealing schedule controlling the convergence of the algorithm to a final configuration).

Figure 4 illustrates the structure of the Bayesian network. The selection of variables constituting the Bayesian network is based on what we found relevant for the user. These variables encapsulate various states, such as “the member’s profession” (profession), “the gender of the member” (gender), “the preferences of a member regarding the displayed documents” (documents), and “the preferences of a member regarding the displayed graphic components” (graphic), among others. The data collected earlier and the established structure of the Bayesian model are employed in the inference algorithm.

# 5 Bayesian Model

In the Bayesian framework, inference involves calculating the probability distribution for all variables based on the given evidence (or set of observations).

## 5.1 The Initial Bayesian Model

After constructing the Bayesian network, it becomes a valuable tool for reasoning about the model's state. Bayesian inference entails calculating the probability distribution across all variables, taking into account the set of observed variables. In the initial phases of our research, we conducted Bayesian inference utilizing the Weka software [54]. Initially, all observed variables in our Bayesian model, including "age," "gender," "focus," and "profession," were treated as hard evidence. Figure 5 depicts the results of Bayesian inference [46, 47].

The original Bayesian network is utilized to deduce user preferences, encompassing choices related to graphical components, advertising, and the selection of displayed documents. After establishing the values for all observed variables, Bayesian inference is employed to forecast user preferences in areas such as advertising and graphics. Variables with higher probabilities are anticipated to exert a more substantial influence on shaping the user interface compared to those with lower probabilities.

## 5.2 Bayesian Model Using Junction Tree Algorithm

The Junction Tree Algorithm is applicable to networks of different types, irrespective of whether they exhibit tree structures. Its core procedure includes transforming the graph into a junction tree. Following this, it initializes potentials, engages in message passing to propagate messages, and calculates posterior probabilities. The algorithm operates as outlined by [24]:

- Graph Transformation: The directed acyclic graph (DAG) of a belief network undergoes transformation into a join tree structure
- Global Propagation: A sequence of ordered local manipulations, referred to as message passes, is executed on the join tree potentials. These message passes restructure the join tree potentials to guarantee local consistency, yielding a consistent join tree.
- Marginalization: Utilizing the consistent join tree, the probability distribution for each variable is computed.

In conventional Bayesian networks, observations usually entail a singular state of the observed node. In simpler terms, when there is evidence in a node X, only one of its states is observed, assigned the value 1, while the remaining states are assigned the value 0. Nevertheless, this method falls short in representing scenarios where observations are uncertain. This uncertainty implies that the observation simultaneously relates to multiple states, each with well-defined percentages.

In our study, we utilize the Junction Tree algorithm to propagate evidence due to its precision and efficiency in handling a restricted number of nodes. Ultimately, the outcomes of this algorithm become apparent after undergoing the following steps: Moralization, Triangulation, and Marginalization.

## 6 Evidence for Bayesian Inference Model

Three types of non-deterministic evidence are identified. Likelihood evidence is defined by its imprecision, reflecting uncertainty and quantified through a likelihood ratio. It is then propagated using Pearl's method of virtual evidence. Probabilistic evidence places constraints on the state of specific variables following propagation in the Bayesian network (BN). Bayesian Networks offer a means to analyze the status of models upon their creation.

Inference algorithms play a crucial role in calculating posterior probability distributions for all variables. In our research, we utilized the Junction Tree algorithm [24], a versatile algorithmic framework that elucidates fundamental concepts in inference. Variables with higher probabilities are presumed to play a more significant role in generating an adaptive user interface compared to those with lower probabilities. Hence, the Bayesian user interface can be tailored based on these probabilities.

Consequently, the Bayesian user model is prepared for predicting user preferences, influencing the display of documents and advertisements based on heightened user preferences. Our web interface is adaptive, providing recommendations to guide users in selecting their preferences. For new members who haven't specified their preferences, the Bayesian user model is employed to establish default values

## 7 Conclusion

In this paper, we introduced our research and proposed an adaptive user interface utilizing the Bayesian network approach. Initially, we conducted a comparison of learning algorithms to train the Bayesian network structure and implemented Bayesian inference into our web interface. Our approach was further validated through an evaluation of the web interface.

Compared to the fixed user interface, our findings indicated that the proposed interface garnered higher user satisfaction than the fixed one. To enhance the model, we incorporated a more robust training approach by utilizing a log of users' activities to specify the Bayesian network. To further improve system evaluation, we plan to incorporate qualitative measures regarding user experience and satisfaction derived from verbal user feedback.

## References

1. Bilmes, J.: On soft evidence in Bayesian networks. UWEE Tech Report Series, Department of Electrical Engineering University of Washington (2004)
2. Bloemeke, M. R.: Agent encapsulated Bayesian networks. PhD Thesis, Department of Computer Science, University of South Carolina (1998)

3. Brusilovsky, P., Millán, E.: User models for adaptive hypermedia and adaptive educational systems. The Adaptive Web. Lecture Notes in Computer Science, vol 4321, pp. 3–53 (2007) doi: 10.1007/978-3-540-72079-9_1
4. Butz, C. J., Fang, F.: Incorporating evidence in Bayesian networks with the select operator. In: Proceedings of the 18th Canadian Conference on Artificial Intelligence. Advances in Artificial Intelligence, vol. 3501, pp. 297–301 (2005) doi: 10.1007/11424918_31
5. Chan, H., Darwiche, A.: On the revision of probabilistic beliefs using uncertain evidence. Artificial Intelligence, vol. 163, no. 1, pp. 67–90 (2005) doi: 10.1016/j.artint.2004.09.005
6. Chan, H., Darwiche, A.: Sensitivity analysis in bayesian networks: From single to multiple parameters. In: Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence, pp. 67–75 (2012) doi: 10.48550/ARXIV.1207.4124
7. Chan, H.: Sensitivity Analysis of Probabilistic Graphical Models. PhD Thesis, University of California (2005)
8. Conati, C., Gertner, A., Vanlehn, K.: Using bayesian networks to manage uncertainty in student modeling. User Modeling and User-Adapted Interaction, vol. 12, no. 4, pp. 371–417 (2002) doi: 10.1023/a:1021258506583
9. Conati, C., Gertner, A., Vanlehn, K.: Using bayesian networks to manage uncertainty in student modeling. User Modeling and User-Adapted Interaction, vol. 12, no. 4, pp. 371–417 (2002) doi: 10.1023/a:1021258506583
10. Darwiche, A. : Modeling and Reasoning with Bayesian Networks. Cambridge University Press (2011) doi: 10.1017/CBO9780511811357
11. Deming, W. E., Stephan, F. F.: On a least square adjustment of a sampled frequency table when the expected marginal totals are known. The Annals of Mathematical Statistics, vol. 11, no. 4, pp. 427–444 (1940)
12. Dey, A. K.: Understanding and using context. Personal and Ubiquitous Computing, vol. 5, no. 1, pp. 4–7 (2001) doi: 10.1007/s007790170019
13. D'Ambrosio, B., Takikawa, M., Upper, D.: Representation for dynamic situation modeling.Bayesian Reading Group Technical report (2000)
14. Fink, J., Kobsa, A.: User modeling for personalized city tours. Artificial Intelligence Review, vol. 18, no. 1, pp. 33–74 (2002) doi: 10.1023/a:1016383418977
15. Fischer, G.: User modeling in human–computer interaction. User Modeling and User-Adapted Interaction, vol. 11, no. 1/2, pp. 65–86 (2001) doi: 10.1023/a:1011145532042
16. Gonzales, C. , Wuillemin, P. H.: Réseaux bayésiens en modélisation d'utilisateurs. Sciences et Technologies de l'Information et de la Communication pour l'Éducation et la Formation, vol. 5, no. 2, pp. 173–198 (1998)
17. Good, N., Ben-Schafer, J., Konstan, J. A., Borchers, A., Sarwar, B., Herlocker, J., Riedl, J.: Combining collaborative filtering with personal agents for better recommendations. In: Proceedings of the 16th National Conference on Artificial Intelligence and 11th Conference on Innovative Applications of Artificial Intelligence, American Association for Artificial Intelligence, pp. 439-446 (1999)
18. Hong, J., Yang, S., Cho, S.: ConaMSN: A context-aware messenger using dynamic bayesian networks with wearable sensors. Expert Systems with Applications, vol. 37, no. 6, pp. 4680–4686 (2010) doi: 10.1016/j.eswa.2009.12.040
19. Horvitz, E. J., Breese, J. S., Heckerman, D., Hovel, D., Rommelse, K.: The lumiere project: Bayesian user modeling for inferring the goals and needs of software users. Bayesian User Modeling, pp. 256–265 (2013) doi: 10.48550/ARXIV.1301.7385
20. Horvitz, E., Kadie, C., Paek, T., Hovel, D.: Models of attention in computing and communication: From principles to applications. Communications of the ACM, vol. 46, no. 3, pp. 52–59 (2003) doi: 10.1145/636772.636798
21. Jeffrey, R. C.: The logic of decision. University of Chicago Press, 2nd ed, pp. 246 (1990)

22. Kim, Y., Valtorta, M., Vomlel, J.: A prototypical system for soft evidential update. Applied Intelligence, vol. 21, no. 1, pp. 81–97 (2004) doi: 10.1023/b:apin.0000027768.02013.54
23. Kjærulff, U. B., Madsen, A. L.: Bayesian networks and influence diagrams: A guide to construction and analysis. Information Science and Statistics, 2nd ed (2013) doi: 10.1007/978-1-4614-5104-4
24. Koller, D., Friedman, N.: Probabilistic graphical models: Principles and techniques. MIT Press (2009)
25. Korb, K. E., Nicholson, A. E.: Bayesian artificial intelligence (Chapman and Hall / CRC Computer Science and Data Analysis). CRC PressINC, 2nd ed (2010)
26. Korpipää, P., Koskinen, M., Peltola, J., Mäkelä, S., Seppänen, T.: Bayesian approach to sensor-based context awareness. Personal and Ubiquitous Computing, vol. 7, no. 2, pp. 113–124 (2003) doi: 10.1007/s00779-003-0237-8
27. Koski, T. , Noble, J. M.: Bayesian networks: An introduction. Wiley Series in Probability and Statistics (2009) doi: 10.1002/9780470684023
28. Krieg, M. L.: A tutorial on Bayesian belief networks. Tech Report Surveillance Systems Division Electronics and Surveillance Research Laboratory (2001)
29. Langevin, S., Valtorta, M., Bloemeke, M.: Agent-encapsulated Bayesian networks and the rumor problem. In: Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems, vol. 1, pp. 1553–1554 (2010)
30. Langevin, S., Valtorta, M.: Performance evaluation of algorithms for soft evidential update in bayesian networks: first results. , pp. 284–297 (2008) doi: 10.1007/978-3-540-87993-0_23
31. Lu, J., Shambour, Q., Xu, Y., Lin, Q., Zhang, G.: A web-based personalized business partner recommendation system using fuzzy semantic techniques. Computational Intelligence, vol. 29, no. 1, pp. 37–69 (2012) doi: 10.1111/j.1467-8640.2012.00427.x
32. Mayo, M. , Mitrovic, A.: Optimising ITS behaviour with Bayesian networks and decision theory. International Journal of Artificial Intelligence in Education, vol. 12, pp. 124–153 (2001)
33. McTear, M. F.: User modelling for adaptive computer systems: a survey of recent developments. Artificial Intelligence Review, vol. 7, no. 3–4, pp. 157–184 (1993) doi: 10.1007/bf00849553
34. Middleton, S. E., Shadbolt, N. R., Roure, D. C. D.: Ontological user profiling in recommender systems. ACM Transactions on Information Systems, vol. 22, no. 1, pp. 54–88 (2004) doi: 10.1145/963770.963773
35. Millán, E., Loboda, T., Pérez-de-la-Cruz, J. L.: Bayesian networks for student model engineering. Computers and Education, vol. 55, no. 4, pp. 1663–1683 (2010) doi: 10.1016/j.compedu.2010.07.010
36. Mooney, R. J., Roy, L.: Content-based book recommending using learning for text categorization. In: Proceedings of the 5th ACM conference on Digital libraries (2000) doi: 10.1145/336597.336662
37. Mrad, A. B., Delcroix, V., Piechowiak, S., Leicester, P., Abid, M.: An explication of uncertain evidence in bayesian networks: Likelihood evidence and probabilistic evidence: Uncertain evidence in bayesian networks. Applied Intelligence, vol. 43, no. 4, pp. 802–824 (2015) doi: 10.1007/s10489-015-0678-6
38. Mrad, A. B., Maalej, M. A., Delcroix, V., Piechowiak, S., Abid, M.: Fuzzy evidence in Bayesian network. In: Proceedings of the International Conference of Soft Computing and Pattern Recognition, pp. 486–491 (2011) doi: 10.1109/socpar.2011.6089143
39. Naïm, P., Wuillemin, P. H., Leray, P., Pourret, O., Becker, A.: Réseaux bayésiens. Eyrolles Group (2007)
40. Nguyen, L., Do, P.: Combination of Bayesian network and overlay model in user modeling. In: Proceedings of the International Conference on Computational Science, pp. 5–14 (2009) doi: 10.1007/978-3-642-01973-9_2

41. Pan, R., Peng, Y., Ding, Z.: Belief update in bayesian networks using uncertain evidence. In: Proceedings of the 18th IEEE International Conference on Tools with Artificial Intelligence, pp. 441–444 (2006) doi: 10.1109/ictai.2006.39
42. Pearl, J.: Probabilistic reasoning in intelligent systems: Networks of plausible inference. Elsevier (1988) doi: 10.1016/c2009-0-27609-4
43. Peng, Y., Ding, Z., Zhang, S., Pan, R.: Bayesian network revision with probabilistic constraints. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, vol. 20, no. 3, pp. 317–337 (2012) doi: 10.1142/s021848851250016x
44. Peng, Y., Zhang, S., Pan, R.: Bayesian network reasoning with uncertain evidences. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, vol. 18, no. 5, pp. 539–564 (2010) doi: 10.1142/s0218488510006696
45. Rich, E.: Stereotypes and user modeling. User Models in Dialog Systems, Symbolic Computation, pp. 35–51 (1989) doi: 10.1007/978-3-642-83230-7_2
46. Rim, R., Amin, M. M., Adel, M.: Bayesian networks for user modeling: predicting the user's preferences. In: Proceedings of the 13th International Conference on Hybrid Intelligent Systems, pp. 144–148 (2013) doi: 10.1109/his.2013.6920472
47. Rim, R., Maalej, M. A., Mahfoudhi, A., Abid, M.: Building and evaluating an adaptive user interface using a bayesian network approach. International Journal of Computer Science and Information Security, vol. 14, pp. 548–565 (2016)
48. Sbattella, L. , Tedesco, R.: Profiling and tutoring users in virtual campus. Proceedings of the 5th International Conference on Information Technology Based Higher Education and Training, pp. 256–262 (2004) doi: 10.1109/ITHET.2004.1358174
49. Song, I., Cho, S.: Bayesian and behavior networks for context-adaptive user interface in a ubiquitous home environment. Expert Systems with Applications, vol. 40, no. 5, pp. 1827–1838 (2013) doi: 10.1016/j.eswa.2012.09.019
50. Tedesco, R., Dolog, P., Nejdl, W., Allert, A.: Distributed Bayesian networks for user modeling. In: Proceedings of the E-Learn, World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education (2006)
51. Tomaso, E. D., Baldwin, J.: An approach to hybrid probabilistic models. International Journal of Approximate Reasoning, vol. 47, no. 2, pp. 202–218 (2008) doi: 10.1016/j.ijar.2007.04.004
52. Valtorta, M., Kim, Y., Vomlel, J.: Soft evidential update for probabilistic multiagent systems. International Journal of Approximate Reasoning, vol. 29, no. 1, pp. 71–106 (2002) doi: 10.1016/s0888-613x(01)00056-1
53. Wei, K., Huang, J., Fu, S.: A survey of e-commerce recommender systems. In: International Conference on Service Systems and Service Management, pp. 1–5 (2007) doi: 10.1109/icsssm.2007.4280214
54. Weka 3: Machine learning software in java (2023) www.cs.waikato.ac.nz/ml/weka/
55. Zhang, S., Peng, Y., Wang, X.: An efficient method for probabilistic knowledge integration. In: Proceedings of the 20th IEEE International Conference on Tools with Artificial Intelligence, pp. 179–182 (2008) doi: 10.1109/ictai.2008.57

Electronic edition
Available online: http://www.rcs.cic.ipn.mx

http://rcs.cic.ipn.mx

Centro de Investigación en Computación