# Thesis Review and Analysis Automated System

Jean Patrick Lostaunau, Armando Soto, Alfredo Barrientos
Universidad Peruana de Ciencias Aplicadas
Lima, Perú
{u20171A843, u201719037, pcsiabar}@upc.edu.pe

*Abstract* — **In this article, we propose the design, construction, and validation of a technological solution with the ability to automate the process of reviewing and analyzing thesis paper by using natural language processing and a deep learning training algorithm. This project seeks to be a proof of concept because current solutions in the academic field focus on abstracting and analyzing scientific articles but not in theses. Another point to consider is that these solutions are in English language. The resulting language model was compared with other language models based on the Transformers architecture. The result of this comparison gives us an objective for future research on the A.S.T.R.A. project.**

## I. INTRODUCTION

In Peru, the number of applicants for Systems Engineering and Software careers in various universities tends to increase since 2015. Universidad Peruana de Ciencias Aplicadas (UPC) also increased the number of its graduates by 24% from 2014 to 2019 in Systems Engineering and Software Engineering's careers [1]. Thesis projects are mostly developed in the last year of a UPC student's career, so if the number of graduates increases, the number of theses that must be developed will increase, therefore, the university will require to increase the number of teachers or a more efficient process over time.

A thesis from the School of Systems engineering and computing contains approximately forty thousand words. The calculation was based on the average of words of fifty theses of the systems engineering career extracted from the thesis repository of the UPC. Considering the number of words and the number of words that a regular lector reads on average from 200 to 400 words in a minute [2] so it will take a teacher from 1h. 40min. to 3h. 20 min. the corresponding reading before the analysis and revision of a thesis document. According to a survey done to teachers who participate in the thesis review process in System and Computer Engineering School of UPC, on average the time of review of a thesis by the evaluators is 9 h. 7 min. That score was calculated using the sum of the average review time in three different stages done in the revision of theses. Finally, this causes the need to develop a solution for automating the process of revision and linguistic analysis of these documents of the school of systems engineering and computing.

## II. RELATED WORK

The information analysis is a task that refers to the detailed evaluation of an investigation to know the different components that make it up to understand the relationship that exists within each of these. [6].

### *A. Existing solutions for automating the thesis review process*

In the literature research carried out, it has been possible to identify a certain number of solutions that speed up the thesis review process. This includes web applications, mobile applications, evaluation rubrics and frameworks. Table I describes the automated solutions that exists at the market for these review process.

TABLE I. MAIN AUTOMATED SOLUTIONS OF THE THESES REVIEW PROCESS

| N° | Solution | Definition | Observation |
|---|---|---|---|
| 1 | Google Classroom [7] | Tool used for classroom activities. | Show document's originality. |
| 2 | Evaluation Rubric [8] | Document that manages the expectation for a course by using a criteria. | Display how to make use of examiners perspective for the overall thesis grade. |
| 3 | Framework [9] | Tool that helps students work forward to examination expectations. | Framework which works to know examination expectations. |
| 4 | Blackboard Collaborate [10] | Tool used for online learning. | Discuss about a plagiarism checker tool. |

It should be noted that the solutions shown above do not have an approach to the analysis of the structure, nor the information found in the thesis document to be used in a context of questions and answers, that is, they only improve other aspects of the thesis review process.

### *B. Linguistic analysis within the review of reports*

The academic field of natural language processing has undergone a positive variation during the last 3 years in which the number of publications covering this topic within which the topics that have the most relevance are recurrent neural networks (60.8%) and word2vec (74.1%) [11].

In the academic field you can find a number of publications that refer to the use of natural language processing tools to analyze a certain information document and obtain a benefit such as, questions and answers, summaries, entity detection and information autocomplete. As a result of the research done, we created TABLE II.

TABLE II. LINGUISTIC ANALYSIS WITHIN THE REVIEW OF REPORTS

| N° | Task | Observation |
|---|---|---|
| 1 | Questions and Answers [12] | Framework to get response from a text in English. It mentioned that it was used only to provide the correct answers regarding history, famous monuments of a city in India. |
| 2 | Summary of Information [14] [15] [16] | Tools that provide a summary of a big amount of text documents. It can be used for other kinds of solutions which require the summary of theses. |
| 3 | Sentences autocomplete [13] | This field generate text from other linguistic structures and need some input as a historic text to generate new sentences. It can be used as a part of other solution, but itself doesn't have relevance for a report's review process. |
| 4 | Feature discovery [11] [12] [19] | The entity recognition task is useful to classify key concepts in the academic field and structures like algorithms, but the review of a report needs a comparison of the content and a rubric establishments. |
| 5 | Plagiarism detection [17] [18] | Plagiarism detection compares the text contained in the report with other reports and identifies similarities, which speeds up the process of reviewing reports, fulfilling its objective. However, the rest of a report review process still represents a significant percentage with respect to the complete review process. |

There are solutions that use the task of questions and answers, however, solutions that involve automated theses review are in English. In the case of the study [12] the approach used by the authors requires an artificial intelligence algorithm and a set of questions and answers in English without the need to use a context for the solution to work correctly.

Plagiarism detection refers to the verification of the similarity between one academic report and another. This is done to protect copyright and prevent academic dishonesty. [20]. This is responsible for analyzing all the information in a document and then comparing it with a database of doctoral theses.

## III. PROPOSED SOLUTION

The proposed solution consists of developing a proof of concept of a technological solution that allows automation in the process of revision and linguistic analysis of thesis documents. The solution addresses the problem by providing an information extraction tool from a thesis document to which a teacher can ask a question in natural language and this tool provides one answer. The proposed solution includes a web application with which teachers manage thesis documents and ask the relevant questions. For the web application to provide an answer to the question posed by the teacher, it was necessary to involve a technology that can extract data by analyzing natural language and interpreting the context provided, which in this case is the final thesis document. Considering this need, the union of two fields of computer science, the field of Natural Language Processing [4] (NLP) and Deep Learning [3] (DL) was considered inspired by Bert model [21], resulting in a DL algorithm that uses NLP concepts and techniques that is commonly called the Language Model.

### *A. Web application*

The web application has been designed based on the current process of reviewing thesis documents of UPC. The solution covers the last section of the thesis review process which involves two roles, the evaluation jury that performs the initial validation of the initial research project to determine a grade and the career coordinator, who performs a second evaluation of the project. In both cases, the final document that encompasses all the research that we will call "memory" and the documents attached to the project are reviewed.

The scope of this project includes the review of the report in PDF format and answering questions that the teacher asks considering this document. In the application you can enter information about users (teachers, coordinators, directors) and projects. A user with a teacher role can be assigned to a project, which performs the function of attaching the memory file in PDF format to ask questions.

In the logical architecture of the system (see Fig. 1) it can be seen that 3 roles interact, the Director, who has administrator permissions and manages the information of the teachers and projects, the Professor, who can attach the memory document to the application and ask questions; and the Coordinator, who can ask questions on all projects that have been created in the database.

The system has three key components, a SPA web application developed with Vue.js with which the user interacts, a Rest API developed in Express.js which handles the business logic and a microservice type API that processes the questions and makes use of the developed Language Model.

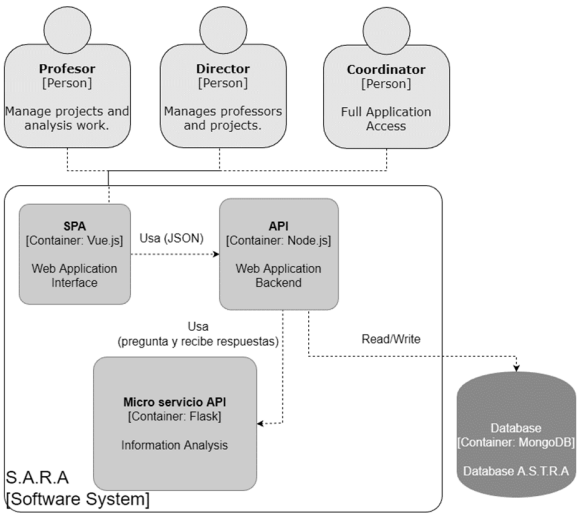



Fig. 1. Logical Architecture of the Web Application

*B. Language model:*

The extraction of information is the main feature of the solution, to perform this task it was necessary to develop a Language Model that fits the way teachers review the memory of a thesis. A first analysis of the thesis review process revealed that a teacher reviews a rubric that the university provides and extracts information from a document locating the section of this where the necessary information for its evaluation is located. It should be noted that each teacher reviews a document in diverse ways, however, what never changes is the need to read the document and locate the necessary information.

The NLP field focuses on "tasks" types such as text generation, text summary, questions and answers, entity identification, among others. An algorithm that can locate, extract information, and formulate a question based on a question needs to perform more than one task (e.g. Identify citations, differentiate academic from colloquial language). That is why we opted for a Deep Learning solution, because each layer can be used for a specific task and change its configuration to optimize the accuracy with which it provides an answer.

A Deep Learning solution is initialized with a random configuration that is adjusted as the training is processed. From this point there are two options, the first is to train a randomly initialized model and the second option is to start from a pre-trained model. Pre-trained models are algorithms that have already been trained with data and are general purpose, that is, they do not specialize in a specific task, they can understand human language at a general level. The option chosen was the second, since it required a smaller amount of data to improve the accuracy of its answers.

An analysis was carried out of the existing language models that have similar characteristics to the proposed solution. First, a search was carried out for models that provide solutions to problems related to the academic area. Then, a filter was made for those models that focused on scientific articles. Due to the large number of existing models it is necessary to prioritize the analysis of these. If we assign a weight to each criterion, we can derive a score to each model and sort the list excluding models that are not relevant to our solution. Results are shown in Table III. The allocation of the weight of the score is distributed as follows:

- General Purpose (1)
- Trained with academic reports (3)
- That support the Spanish language (2)

TABLE III. COMPARISON OF LANGUAGE MODELS

| Model | Total Value |
|---|---|
| Sci-Bert | 4 |
| SpERT | 3 |
| MarianMT | 2 |
| Bert | 1 |
| BigBird | 1 |
| CamenBERT | 1 |
| ConvBERT | 1 |
| DeBERTa | 1 |
| DeBERTa-v2 | 1 |
| DPR | 1 |
| Flaubert | 1 |
| Funnel Transformer | 1 |
| I - BERT | 1 |
| LED | 1 |
| Longformer | 1 |
| LXMERT | 1 |
| MobileBERT | 1 |
| MPnet | 1 |
| Reformer | 1 |
| Roberta | 1 |
| SqueezeBERT | 1 |
| Q5 | 1 |
| TAPAS | 1 |
| XLM | 1 |
| XLM - RoBERTa | 1 |
| XLNet | 1 |

There are some models that are the product of applying the fine-tuning technique and then performing a second training with new data (SciBert, SpERT). These models are called retrained models and when you want to refer to the initial model (before fine-tuning) they are called distributions (e.g. SciBERT is a BERT distribution). Prioritization over the models chosen after the filter reduced the possible candidate models to SciBERT [22] and SpERT [23] The models meet the established requirements, were trained with scientific articles and are BERT distributions, which means that they are derived from a general purpose model, which facilitates the retraining of both models. None meet the criteria of being in Spanish, but BERT has distributions that work multilingually, so using methods to get one of these models to accept the Spanish language is feasible. The main problem of both models is that they were trained in English, this lies in a problem when coding the words, since the element that encodes them to numerical matrices, the encoder, also requires training. However, the matrix architecture (Transformers) does not change in both cases, only its configuration. The methodology that SciBERT used in its paper was to train with 1.14 million scientific articles and SpERT had two supervised trainings of identification of entities. Due to the limitations of the English language in both models, we chose to use BERT in a version that was trained with Spanish text and use SciBERT's idea of unsupervised training. SpERT used a supervised training in two stages, one in which it was given a general understanding of the entities to be recognized, and another where the task to be performed was

specified. Even so, these models cannot be candidates for retraining due to their encoder in English, they gave us an understanding of how to propose a strategy to create a model that solves the problem posed and adapts to the casuistry of the proposed solution.

Considering the previous analysis of evaluated models, it was determined that for the training it was necessary to train the algorithm in two stages. The first was under the paradigm of unsupervised training. This will be carried out with two hundred theses from the Faculty of Engineering of the UPC. The second stage will be under the paradigm of supervised training providing the model with a set of questions with their respective answers for the algorithm to learn to answer various questions related to the information of a thesis applying the Fine-tuning technique.

### *C. Unsupervised training:*

The unsupervised training required training the model with blocks of text that are part of thesis papers. The data set had to contain these blocks of text that are part of a thesis correctly ordered and separated with an identifier to be processed. The stages for unsupervised training are described below:

#### *C.1. Massive download of thesis*

In this first stage, a robot was created using Uipath using a process automation tool with robots to be able to perform Web Scraping of the repository of the Universidad Peruana de Ciencias Aplicadas (https://repositorioacademico.upc.edu.pe/) to download a massive number of theses and thus build the data set satisfactorily. This is the flow that the robot will follow to download the theses (see Fig. 2):

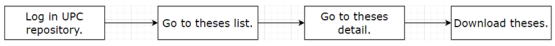



Fig. 2. Operation of the robot that downloads the theses

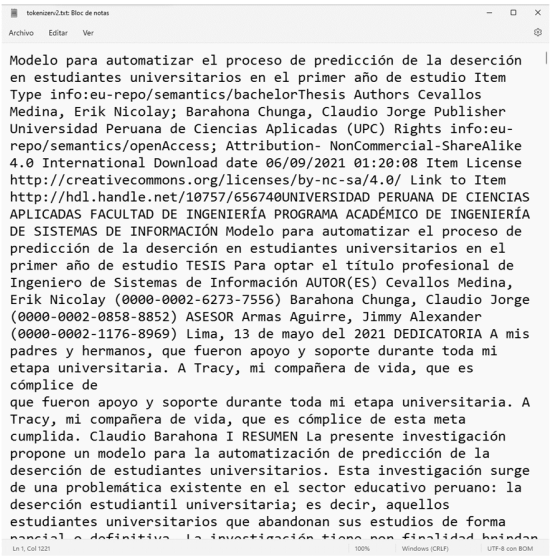
The robot downloaded an approximate of two hundred theses in PDF format of the Careers of Systems Engineering and Software Engineering. The main problem now focused on how to extract the text from pdf files and transform it into a single file that meets the requirements for training.

#### *C.2. Data transformation*

In the second stage, another robot was developed for extracting text from PDF files and transforming it into a plain text file. Fig. 3 shows the initial status of the folder that contained the PDF files.

Fig. 3. Bot Result

The robot went through each PDF file and retrieved the text that was inside it, additionally the robot cataloged blocks of text of two hundred words and included a separator every time the number of words in the current counter is two hundred, when that happened, the counter was 0 again and started the process again. For each iteration, each retrieved record is stored in a variable including an identifier at the end of each iteration. The result can be displayed in Fig. 4 which includes in a plain text file resulting from the execution of the robot. The text file got 44415-word blocks and each block got 200 hundred words approximately.


tokenizer2.txt: Bloc de notas

Archivo Editar Ver

Modelo para automatizar el proceso de predicción de la deserción en estudiantes universitarios en el primer año de estudio Item Type info:eu-repo/semantics/bachelorThesis Authors Cevallos Medina, Erik Nicolay; Barahona Chunga, Claudio Jorge Publisher Universidad Peruana de Ciencias Aplicadas (UPC) Rights info:eu-repo/semantics/openAccess; Attribution- NonCommercial-ShareAlike 4.0 International Download date 06/09/2021 01:20:08 Item License http://creativecommons.org/licenses/by-nc-sa/4.0/ Link to Item http://hdl.handle.net/10757/656740UNIVERSIDAD PERUANA DE CIENCIAS APLICADAS FACULTAD DE INGENIERÍA PROGRAMA ACADÉMICO DE INGENIERÍA DE SISTEMAS DE INFORMACIÓN Modelo para automatizar el proceso de predicción de la deserción en estudiantes universitarios en el primer año de estudio TESIS Para optar el título profesional de Ingeniero de Sistemas de Información AUTOR(ES) Cevallos Medina, Erik Nicolay (0000-0002-6273-7556) Barahona Chunga, Claudio Jorge (0000-0002-0858-8852) ASESOR Armas Aguirre, Jimmy Alexander (0000-0002-1176-8969) Lima, 13 de mayo del 2021 DEDICATORIA A mis padres y hermanos, que fueron apoyo y soporte durante toda mi etapa universitaria. A Tracy, mi compañera de vida, que es cómplice de
que fueron apoyo y soporte durante toda mi etapa universitaria. A Tracy, mi compañera de vida, que es cómplice de esta meta cumplida. Claudio Barahona I RESUMEN La presente investigación propone un modelo para la automatización de predicción de la deserción de estudiantes universitarios. Esta investigación surge de una problemática existente en el sector educativo peruano: la deserción estudiantil universitaria; es decir, aquellos estudiantes universitarios que abandonan sus estudios de forma

Ln 1, Col 1221 100% Windows (CRLF) UTF-8 con BOM

Fig. 4. Plain text file resulting from the execution of the data transformation robot

#### *C.3. Development of the algorithm of unsupervised training*

The elaboration of the algorithm for the unsupervised training was carried out in Python using the Transformers library of the Hugging Face (https://huggingface.co/) organization. The library provided us with the necessary tools for a standard training algorithm. The required configuration is extracted from the chosen model (BERT in Spanish) which has the name "bert-base-spanish-wwm-uncased" within the Hugging Face model repository. The configuration of the model was the standard that gave us the library, which is in the BertConfig object of the same. The processing of the data set was performed following the practices described on the Hugging Face website. Using the LineByLineTextDataset format, the plain text file was processed recognizing the identifier that the robot placed every two hundred words. The training of the model was carried out through the Trainer function that gives us the Transformers library that receives as a parameter a TrainingArguments object that contains the initial configuration of BERT, its encoder and the training data set. As a part of the configuration, we considered 20 epochs and a total of 200 steps for the analysis. The data set was separated into 80% training and 20% validation. The training that was done is partly supervised, since internally what Transformers provided us was to mask words randomly before the training and identify the blocks of text sent as part ofthe data set. By identifying these

blocks of text, the model predicted what the next sequence of text was and compared it with the one found in the data set. This is how an unsupervised training is simulated by letting the model make the predictions and compare them with the next block of text that follows in each iteration. An important fact to highlight is on the Y axis has a natural range from 0 to 5 and not from 0 to 2.5 as seen in the image. Remember that we start with the Spanish model of BERT, so it is natural that the training starts better than one that starts randomly.

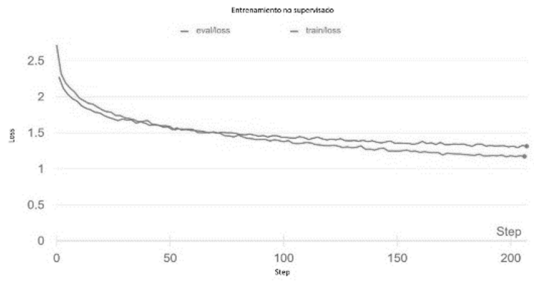



Fig. 5. Results of unsupervised training.

Fig. 5 shows the results of unsupervised training based on approximately two hundred steps, one for each record. Ideally, both lines (Train and Loss) should continue to descend with respect to the Y axis (Loss) until they reach close to zero as the steps increase. It can be observed that within the range established on the Y axis (0-2.5) the Loss Error does not descend from one, so it can be deduced that this model can still be trained with more data without the model adjusting too much to the data and produce Overfitting.

Bert was retrained with thesis data from the academic repository of the Universidad Peruana de Ciencias Aplicadas (UPC) and obtained a general purpose language model adapted to the academic language of a thesis document.

*D. Supervised training*

For the supervised training, we chose to use the Fine-Tuning [5] technique, which allows a model to learn to perform a specific task through examples of the task. For our case it was necessary to show him examples of correct questions and answers for his learning. The first step was to elaborate an example dataset and then run the algorithm generated for unsupervised training with some modifications to the parameters.

*D.1. Data set of questions and answers*

The elaboration of this data set was with the help of a "Heavy User" of the thesis review process, a professor from the UPC who provided us with a set of questions that can serve as an example for the extraction of information to a thesis document. The theses downloaded in PDF format were used to elaborate answers to the questions posed by the teacher. The data were formulated in an Excel with the fields "Context", "Question", "Answer", "Start Character", "Final Character", "Total characters" and "Words" (see Fig. 6) being the context, the extract of the thesis where the answer is located, the starting character where the literal answer begins with respect to the context provided and the final character where the answer ends. The data set contains 400 rows.

| Contexto | Pregunta | Respuesta | Carácter de Inicio | Cárácter Final | Total Caracte | Palabras |
|---|---|---|---|---|---|---|
| proyecto, se obtuvieron las siguientes conclusiones, de acuerdo con los objetivos planteados: Acerca del desarrollo de la | ¿Cuáles son las conclusiones ? | propuesta, se concluye que, a través de la selección de componentes tecnológicos obtenidos de la comparación realizada en el primer | 133 | 1164 | 1164 | 162 |
| En síntesis, luego de desarrollar el proyecto, se obtuvieron las siguientes conclusiones, de acuerdo con los objetivos planteados: | ¿Cuáles son las conclusiones ? | Respecto al análisis realizado, se concluye que sí existen arquitecturas tecnológicas para Smart Buildings que utilizan tecnología IoT, tanto | 132 | 1270 | 1270 | 168 |
| proyecto, se obtuvieron las siguientes conclusiones, de acuerdo con los objetivos planteados: | ¿Cuáles son las conclusiones ? | validación, se consiguió corroborar el cumplimiento de requerimientos de la arquitectura propuesta a | 132 | 905 | 905 | 130 |
| objetivo definir y diseñar una | | Buildings con IoT | | | | |

Fig. 6. Data set of questions and answers

*D.2. Fine-tuning*

The algorithm for the execution of Fine-Tuning is like that of unsupervised training, using the "pandas" library we manage to convert the Excel file provided as input into a set of data that the Transformers library can take as a training parameter in the Train function. The modified parameters are "prediction_loss_only=False" and "per_device_eval_batch_size=16". Another important change is in the data set, separating the dataset into training, evaluation, and testing data.

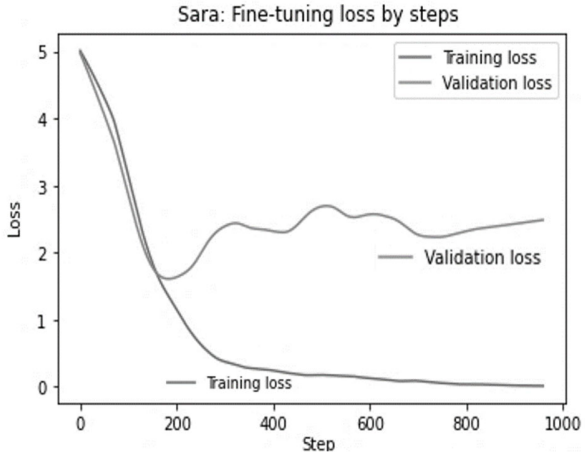



Fig. 7. Distribution of error loss in the validation and training data set

Overfitting is a modeling error in statistics that occurs when a function is too aligned with a limited set of data. Therefore, trying to make the model too close to slightly inaccurate data can infect the model with substantial errors and reduce its predictive power. In Fig. 7 the approximate point where the model stops learning to predict new data (Validation Loss) is in step 200. Evaluating only the section from step 0 to 200, the step can be observed more accurately before the turning point of the Validation Loss, which reflects a overfitting. Fig. 8 shows that the specific point is step 150 before the loss of the validation error begins to rise. The steps function as a checkpoint of the internal configuration of the model, so the 150 is the one that can best predict based on new data. The final

model arises from extracting the model from checkpoint 150 and saving it to the Hugging Face repository.

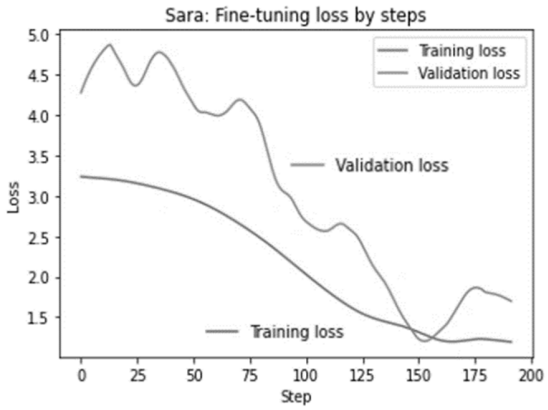



Fig. 8. Fine-tuning result from step 0 to 200

## IV. RESULTS

The artifacts created as part of a solution code called A.S.T.R.A (Automated System for Thesis Review and Analysis) because of the research are described below.

### *A. Artifacts*

#### *A.1. Web application*

The main deliverable of this project is a web application called which constitutes three main components deployed on AWS:

- Front-End (Single Page Application)
- Back-End (API Rest)
- Deep Learning Model (Microservice that allows interaction with the model)

The application, according to the proposed architecture covers the current process of thesis review in its last stage and covers both the management of the professors who are going to review the theses and the projects where these documents can be stored for review.

#### *A.2. Deep learning algorithm*

The creation of the first version of the algorithm is the result of combining other algorithms and systems that allowed us to execute the steps that are followed to reach the development of this artifact.

The requirements arose during the development of the research and the creation of this artifact. The deliverables were built out of necessity as part of the research. In Table IV shows what kind of deliverables are needed by requirement.

TABLE IV. DELIVERABLES BY REQUIREMENT

| Requirement | Action | Deliverable |
|---|---|---|
| Data for model training. | Create an algorithm to obtain data from the academic repository of the UPC (Memories in PDF format). | RPA algorithm – Web Scraping. |
| Transform PDF to text. | Create an algorithm to transform the PDF to text to a file .txt | RPA algorithm – Text extraction from PDF. |
| Data set for training | Transform the data with a Python script including an identifier character different from the rest of the characters in the thesis document. | Data set for training. |
| Training algorithm | Development of an algorithm using a suitable library to perform the training. | Algorithm to train the Deep Learning model. |
| Algorithm that reconfigures the model to answer questions | Develop an algorithm to apply the Fine-Tuning technique and teach the model to answer questions provided by a teacher. | Algorithm to perform the Fine-tuning technique. |
| Data to apply the Fine-Tuning technique | Create a data set in Excel format that includes an excerpt of the thesis, a question, and its respective answer. | Data set in Excel format with examples of questions and answers. |
| Final Algorithm A.S.T.R.A. | Re-train the BERT model with the first data set generated. Then, apply the Fine-tuning technique with the data set of questions and answers. | Deep Learning algorithm that answers questions from teachers. |

In summary, a total of seven deliverables have been developed including the final algorithm used in the proposed web application.

### *B. Validation of experts*

A validation was carried out with an expert who would evaluate our solution proposal based on their work experience in the company that we defined for Our case study (Universidad Peruana de Ciencias Aplicadas) and their vast knowledge in Artificial Intelligence. After the meeting we collected the main comments and suggestions you gave us, which are detailed below.

#### *B.1. Feedback*

The expert mentioned that the project has a good scientific contribution because it is a pioneering solution in its field and describes the first steps to reach a final product in the future.

Regarding the accuracy provided by the Language Model, he mentioned that it is valid that this is low (Less than 4%)

because the model had to process a thesis distributed in blocks of approximately 200 words. An important comment that the expert mentioned is that, due to the nature of this proof of concept, it is normal that the accuracy of the Deep Learning model was reduced, however, this does not mean that in future research the accuracy of the model cannot be increased. It was also mentioned that the focus of the project was to create a first version of the model but not to improve efficiency.

A crucial point that was discovered during the development of the project is that, according to the research carried out in the state of the art and specific objective 1 (Analysis of technologies related to Natural Language Processing), the existing technological solutions focus on scientific papers but not on theses. It is worth mentioning that existing technological solutions are mostly trained with data in English. The above reinforces the fact that this project is a pioneering solution never explored in the field of Deep Learning and Natural Language Processing applied to the field of academic review of higher education.

*B.2. Suggestions*

The expert recommended increasing the volume of data used for initial training by retrieving theses from other repositories and not just theses from the School of Systems and Computer Engineering. It is necessary to consider that the theses that are used for training should not be quite different from each other, otherwise it will be necessary to use more data for a performance that is at the level of a final product.

To improve the accuracy of the model, the expert recommended us to make 5 copies of our initial training set and intersperse the data. Subsequently, perform five workouts with the resulting data sets, compare the accuracy of each of the resulting models and choose the one that obtains the highest precision after doing a test with respect to an evaluation data set.

It was recommended to benchmark other direct access models that are references in the field of Natural Language Processing, as well as to include models in English. This benchmarking will offer a quantitative comparison with respect to other projects. It should be noted that this comparison with other models is to establish a starting point and a goal for future projects that continue with the research.

*C. Quantitative results*

A comparison was made between various language models specialized in the questión-Answering task (Answering questions provided based on a context). The data set that was used was different from the training and validation dataset used to train the A.S.T.R.A. model, averaging the results obtained based on two metrics, F1 Score and Exact Match the results obtained are in Table V. The models used were extracted from the Hugging Face site, ModelHub where you can find more than one thousand model variants.

It is important to note that the use of the F1 score, which combines the measurement of 2 measures, the first, "Recall", which is described as the data extraction score and the second, "Precision" which indicates how close it was to obtaining a similarity with the structure generated from the training data is of vital importance within the development of the project since this will be one of the indicators to measure the performance of the present training [17].

$$F_1 = 2 * \frac{precision * recall}{precision + recall}$$

The Exact Match metric measures the percentage of predictions that exactly match any of the answers. Predictions consider the context provided, if the context contains all the tokens of the answer provided, the metric is 1. On the other hand, the lower the Exact Match, it can be said that it has elaborated a different response to that of the context. This can be interpreted in two ways if we consider the F1 Score, the first is that, if the F1 score is low, and the Exact match as well, the model may be inefficiently predicting a response and does not consider the context to elaborate an answer. The second way to interpret is if the F1 Score is high and the Exact match is low, in that case the model poses a coherent response and is not using the context literally but elaborates a new answer which shows that the model infers and provides a better elaborated response without copying the context.

TABLE V. COMPARISON OF F1 SCORE AND EXACT MATCH BETWEEN A.S.T.R.A. AND OTHER MODELS

| Model | Variant | F1 Score | Exact Match |
|---|---|---|---|
| Roberta | roberta-base-bne-sqac: | 0.2356 | 0.0414 |
| BERT | distill-bert-base-spanish-wwm-cased-finetuned-spa-squad2-en | 0.1800 | 0.0082 |
| BERT | bert-base-spanish-wwm-cased-finetuned-spa-squad2-en | 0.1990 | 0.0207 |
| BERT | bert-base-english-wwm-uncased | 0.1351 | 0.0082 |
| SciBERT (English) | SciBERT-SQuAD-QuAC | 0.1417 | 0.0248 |
| BERT (English) | distilbert-base-cased-distilled-squad | 0.0992 | 0.0248 |
| A.S.T.R.A. | A.S.T.R.A. Question-Answering | 0.1351 | 0.0082 |

## V. CONCLUSIONS

A total of six variants were evaluated from a total of four models (RoBERTa, BERT Spanish, SciBErt-English. BERT-english) and the A.S.T.R.A. model with the variant adjusted to questions and answers. The main comparison is defined based on the Spanish models and the English models. The F1-Score metric is considered since the Exact Match is similar in both cases with a score of 0.0248 which indicates that the English models are not using the same tokens of the provided context, since the Spanish context has a different token encoding only some characters and words can be used in the answers

formulated. However, this does not mean that the answers are provided in English, it only means that the accuracy of the answer will be significantly affected. This is not fully reflected in the results of Scibert (0.1417) and BERT (0.0992) with SciBERT standing out above both. The result of SciBERT can be conditioned to be higher because the academic context is similar, while the evaluation is based on thesis, SciBERT was trained with 1.14 million scientific articles.

The second comparison is the rest of the models with A.S.T.R.A. based on both metrics (F1 and Exact Match). Although A.S.T.R.A. is a trained model with a reduced amount of data it closely resembles the original BERT model "bert-base-spanish-wwm-uncased" and only differs in thousandths. This is because the basis that was used to create A.S.T.R.A. is the standard version of BERT applying re-training and fine-tuning. The most important fact to highlight is that A.S.T.R.A. is the model with the lowest score in F1-Score, however, this does not mean that A.S.T.R.A. (0.1351) is a bad model, it only means that we have a starting point and a goal to aim for, which is RoBERTa (0.2356), a model that uses part of the BERT algorithm, but with different configurations of its parameters. Regarding the next steps of A.S.T.R.A. it is necessary to investigate RoBERTa since it is the model with the highest F1 score and apply other training techniques to improve the accuracy of A.S.T.R.A. Finally, the A.S.T.R.A. model would only lack 0.1105 in F1 score to be a model that can be useful in a context at the market level which is a clear and concise goal that must be considered in future research in A.S.T.R.A.

## VI. FUTURE WORK

Future research refers to a set of projects that can be undertaken to access a new step within research related to the use of natural language processing and deep learning. Future research is divided into stages:

### *A. Thesis extraction using an automated algorithm*

Information extraction is based on extracting computer-readable documents on either a local disk or a cloud repository. The use of an automated algorithm will ensure that this can be automated, thus achieving a reduction in time, costs and human errors when downloading these thesis documents. In this project, automated algorithms have been developed with the ability to massively download the Theses of Information Systems Engineering and Software Engineering from the UPC Repositories website, which is why the focus of this stage is to replicate the previous work but based on other external sources of the university repository.

It is worth mentioning that one of the characteristics required for the construction of the dataset is that the theses are not different from each other, that is, to increase or maintain accuracy it is necessary that the theses are focused on the same field. For example, do not include theses of Administration or Accounting careers along with the careers of Systems Engineering since they are different from each other both in structure and content.

In the current market, there are a large number of tools focused on the automation of processes through robots such as Automation Anywhere, Uipath, Rocketbot, Blues Prism, among others that can perform this task, that is, to download massive thesis files from web repositories.

### *B. Algorithm that supports the analysis of 40,000 words in parallel*

In the development stage of the project was the construction of the unsupervised algorithm that was responsible for training based on a certain number of words per record or line of text. This amount was determined based on the documentation of the Hugging Faces architecture which mentions that the maximum amount was 250 tokens. This translates to an amount of two hundred words per record.

As a more precise system is required, one of the main solutions refers to an algorithm with the ability to support a greater number of tokens and consequently support a greater number of words per data record.

In this study, an analysis of the documents extracted from the academic repository of the UPC was carried out and it was determined that a thesis presents an amount of 40,000 words on average. That is to say that for the correct implementation of this system it is necessary an algorithm with the ability to support and analyze more than 40, 000 words in parallel. That is, an algorithm that tolerates an approximate of 50,000 tokens.

### *C. Elimination of unnecessary elements in thesis documents*

According to the analysis of the thesis documents downloaded using the automated algorithm, it was determined that there were elements that hindered the accuracy of the system's training algorithm. These elements have been classified and are as follows:

- Stalemate.
- Imagery.
- Headlines.
- Footer.
- Indexes.
- Table indexes.
- Indexes of figures.
- Thanks.
- Cover
- Imagery

The approach of this proposal is based on finding a way to eliminate all unnecessary elements either by using an algorithm, automated technologies or a system that has this functionality. In this way it is chosen to have clearer and more concise information so that it can be used within the training.

### *D. Analyze new technological solutions based on natural language processing to automate the thesis review process*

In the present project an analysis of technological solutions based on natural language processing was carried out to automate the thesis review process and as a conclusion it was

determined that "Sci-bert", which is a variant of "Bert", with four points was the model that was going to be used to train the algorithm. In addition, it was determined that by means of "Sci-bert" an unsupervised training algorithm can be trained.

It is worth mentioning that the architecture used is the "Transformers Architecture," but it is not the only one that can be used to design and build the technological solution. The Transformers architecture is based on "Bert" which is the reference for each natural language processing project and is also used by Google.

At this stage it is proposed to redefine and revalidate the solutions that are currently found for the task of analysis and processing of a thesis document to determine a choice that has a higher index of accuracy when analyzing the context based on a question and thus achieve a better answer.

## ACKNOWLEDGMENTS

We would like to thank the professors for their support from the beginning of the project to the end of it. Also, for the university, which gave us the knowledge to grow professionally.

## REFERENCES

[1] Upc. (2020). Applicants, entrants and graduates Undergraduate | Universidad Peruana de Ciencias Aplicadas - UPC. https://www.upc.edu.pe/transparencia-upc/postulantes-ingresantes-y-egresados-pregrado/

[2] Rayner, K., Schotter, E. R., Masson, M. E. J., Potter, M. C., & Treiman, R. (2016). So much to read, so little time: How do we read, and can speed reading help? In Psychological Science in the Public Interest, Supplement (Vol. 17, Issue 1). https://doi.org/10.1177/1529100615623267

[3] Chassagnon, G., Vakalopolou, M., Paragios, N., & Revel, M. P. (2020). Deep learning: definition and perspectives for thoracic imaging. European Radiology, 30(4), 2021–2030. https://doi.org/10.1007/s00330-019-06564-3

[4] Young, T., Hazarika, D., Poria, S., & Cambria, E. (2018). Recent trends in deep learning based natural language processing. In IEEE Computational Intelligence Magazine (Vol. 13, Issue 3, pp. 55–75). Institute of Electrical and Electronics Engineers Inc. https://doi.org/10.1109/MCI.2018.2840738

[5] Vrbancic, G., & Podgorelec, V. (2020). Transfer Learning with Adaptive Fine-Tuning. IEEE Access, 8, 196197–196211. https://doi.org/10.1109/access.2020.3034343

[6] R. (2020, June 25). | analysis Dictionary of the Spanish language (2001). Retrieved from https://www.rae.es/drae2001/an%C3%A1lisis

[7] Perrotta, C., Gulson, K. N., Williamson, B., & Witzenberger, K. (2020). Automation, APIs, and the distributed labour of platform pedagogies in Google Classroom. Critical Studies in Education, 62(1), 97–113. https://doi.org/10.1080/17508487.2020.1855597

[8] Haagsman, M., Snoek, B., Peeters, A., Scager, K., Prins, F., & van Zanten, M. (2021). Examiners' use of rubric criteria for grading bachelor theses. Assessment & Evaluation in Higher Education, 46(8), 1269–1284. https://doi.org/10.1080/02602938.2020.1864287

[9] Hodgson, D. (2017). Helping doctoral students understand PhD thesis examination expectations: A framework and a tool for supervision. Active Learning in Higher Education, 21(1), 51–63. https://doi.org/10.1177/1469787417742020

[10] Hill, L. (2019). Blackboard Collaborate Ultra: An Online, Interactive Teaching Tool. Academy of Management Learning & Education, 18(4), 640–642. https://doi.org/10.5465/amle.2019.0027

[11] Moon, S., Lee, G., Chi, S., & Oh, H. (2021). Automated Construction Specification Review with Named Entity Recognition Using Natural Language Processing. Journal of Construction Engineering and Management, 147(1), 04020147. https://doi.org/10.1061/(asce)co.1943-7862.0001953

[12] Badugu, S., & Manivannan, R. (2020). A study on different closed domain question answering approaches. International Journal of Speech Technology, 23(2), 315–325. https://doi.org/10.1007/s10772-020-09692-0

[13] Wang, H. C., Hsiao, W. C., & Chang, S. H. (2020). Automatic paper writing based on a RNN and the TextRank algorithm. Applied Soft Computing, 97, 106767. https://doi.org/10.1016/j.asoc.2020.106767

[14] Zerva, C., Nghiem, M. Q., Nguyen, N. T. H., & Ananiadou, S. (2020). Cited text span identification for scientific summarisation using pre-trained encoders. Scientometrics, 125(3), 3109–3137. https://doi.org/10.1007/s11192-020-03455-z

[15] Cagliero, L., Garza, P., & Baralis, E. (2019). ELSA. ACM Transactions on Information Systems, 37(2), 1–33. https://doi.org/10.1145/3298987

[16] La Quatra, M., Cagliero, L., & Baralis, E. (2020). Exploiting pivot words to classify and summarize discourse facets of scientific papers. Scientometrics, 125(3), 3139–3157. https://doi.org/10.1007/s11192-020-03532-3

[17] Sánchez-Vega, F., Villatoro-Tello, E., Montes-y-Gómez, M., Rosso, P., Stamatatos, E., & Villaseñor-Pineda, L. (2017). Paraphrase plagiarism identification with character-level features. Pattern Analysis and Applications, 22(2), 669–681. https://doi.org/10.1007/s10044-017-0674-z

[18] Shang, H. F. (2018). An investigation of plagiarism software use and awareness training on English as a foreign language (EFL) students. Journal of Computing in Higher Education, 31(1), 105–120. https://doi.org/10.1007/s12528-018-9193-1

[19] Wang, Y., & Zhang, C. (2020). Using the full-text content of academic articles to identify and evaluate algorithm entities in the domain of natural language processing. Journal of Informetrics, 14(4), 101091. https://doi.org/10.1016/j.joi.2020.101091

[20] Education with integrity. (2021). Retrieved from https://www.turnitin.com/es

[21] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. Advances in Neural Information Processing Systems, 2017-Decem, 5999–6009. https://arxiv.org/abs/1706.03762v5

[22] Beltagy, I., Lo, K., & Cohan, A. (2020). SCIBERT: A pretrained language model for scientific text. EMNLP-IJCNLP 2019 - 2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing, Proceedings of the Conference, 3615–3620. https://doi.org/10.18653/v1/d19-1371

[23] Eberts, M., & Ulges, A. (2020). Span-based joint entity and relation extraction with transformer pre-training. Frontiers in Artificial Intelligence and Applications, 325, 2006–2013. https://doi.org/10.3233/FAIA200321