mathematics

MDPI

Article

Short Answer Detection for Open Questions: A Sequence Labeling Approach with Deep Learning Models

Samuel González-López [1,†], Zeltzyn Guadalupe Montes-Rosales [2,†], Adrián Pastor López-Monroy [2,*,†], Aurelio López-López [3,†] and Jesús Miguel García-Gorrostieta [4,†]

[1] Department of Computer Science, Universidad Tecnológica de Nogales, Nogales 84094, Mexico; sgonzalez@utnogales.edu.mx
[2] Department of Computer Science, Mathematics Research Center (CIMAT), Jalisco s/n, Valenciana, Guanajuato 36023, Mexico; zeltzyn.montes@cimat.mx
[3] Computational Sciences Department, Instituto Nacional de Astrofísica, Óptica y Electrónica, Sta. María Tonantzintla, Puebla 72840, México; allopez@inaoep.mx
[4] Department of Computer Science, Universidad de la Sierra, Moctezuma 84560, Mexico; jgarcia@unisierra.edu.mx
* Correspondence: pastor.lopez@cimat.mx
† These authors contributed equally to this work.

Abstract: Evaluating the response to open questions is a complex process since it requires prior knowledge of a specific topic and language. The computational challenge is to analyze the text by learning from a set of correct examples to train a model and then predict unseen cases. Thus, we will be able to capture patterns that characterize answers to open questions. In this work, we used a sequence labeling and deep learning approach to detect if a text segment corresponds to the answer to an open question. We focused our efforts on analyzing the general objective of a thesis according to three methodological questions: Q1: What will be done? Q2: Why is it going to be done? Q3: How is it going to be done? First, we use the Beginning-Inside-Outside (BIO) format to label a corpus of targets with the help of two annotators. Subsequently, we adapted four state-of-the-art architectures to analyze the objective: Bidirectional Encoder Representations from Transformers (BERT-BETO) for Spanish, Code Switching Embeddings from Language Model (CS-ELMo), Multitask Neural Network (MTNN), and Bidirectional Long Short-Term Memory (Bi-LSTM). The results of the F-measure for detection of the answers to the three questions indicate that the BERT-BETO and CS-ELMo architecture obtained the best effectivity. The architecture that obtained the best results was BERT-BETO. BERT was the architecture that obtained more accurate results. The result of a detection analysis for Q1, Q2 and Q3 on a non-annotated corpus at the graduate and undergraduate levels is also reported. We found that for detecting the three questions, only the doctoral academic level reached 100%; that is, the doctoral objectives did contain the answer to the three questions.

Keywords: question answering; open questions; academic document analysis; sequence labeling; deep learning

MSC: 68T50

Citation: González-López, S.; Montes-Rosales, Z.G.; López-Monroy, A.P.; López-López, A.; García-Gorrostieta, J.M. Short Answer Detection for Open Questions: A Sequence Labeling Approach with Deep Learning Models. *Mathematics* **2022**, *10*, 2259. https://doi.org/10.3390/math10132259

Academic Editors: Nebojsa Bacanin and Catalin Stoean

Received: 26 May 2022
Accepted: 22 June 2022
Published: 28 June 2022

1. Introduction

Writing academic documents such as an essay, a research protocol, or a thesis usually starts when we write the first sentence, then a paragraph, and finally the entire document [1]. However, as we advance in writing, our ideas are connected until we achieve a coherent and structured document. After some experience, we can identify certain patterns or template sequences within the document [2]. For example, during the process of preparing the document, the student tries to answer questions and then writes the ideas and results. When the document is finished, this is usually reviewed by an instructor or academic advisor.

In our work, we focused on searching sequence or text patterns that are key to answering open questions. Specifically, we focused on the detection of answers within the general objective of a thesis at graduate and undergraduate levels, comparing different deep learning architectures under a sequence-labeling scenario.

The general objective of a thesis is one of the most important guides of the investigation and must be present throughout the development of the thesis project. The objective must be specific, measurable, and achievable. Commonly, the objective begins to be written out with a verb in the infinitive, as indicated by the institutional guidelines. For instance, we identified that in areas related to computer science, the objectives include answers to three methodological questions; the first answer indicates the scope of the thesis from a verb in the infinitive form; the second describes the problem to be solved; and the third answer details the method, technique, or tools to be employed. In this paper we use our proposed corpus, with the aim of training and testing the models. Thus, we define Q1, Q2 and Q3 as the following open questions:

Q1: *What is going to be done?* Object to be achieved

Q2: *Why is going to be done?* The main purpose of the object

Q3: *How is it going to be done?* Activities or instruments to achieve the object

Below we show an example of the segments identified for a general objective of a thesis.

Objective: *Design an architecture that allows the control and monitoring of various robots simultaneously through the cloud and serverless tools.*

Q1: *Design an architecture*

Q2: *... allows the control and monitoring of various robots simultaneously*

Q3: *through the cloud and serverless tools*

We want to remark that the methodological questions are open, i.e., the student usually has the freedom to write the specific topic of the objective. Each answer's length can vary; in particular, in our corpus, the average number of words used by students for Q1 is 8 with a standard deviation (SD) of 4; that is, the answers for Q1 can contain between 4 and 12 words. Unlike question answering, which mainly focuses on factual answers (what, why, who, how, where questions) [3], the answers to the questions of the addressed problem are made up of several words. We can observe a high variability in Q2 and Q3, which makes the task complex. For Q2, the average number is 13 with an SD of 6, and Q3 contains 9 words with an SD of 6 words. Thus one of our goals was to focus on capturing the structure of the answers for each open question.

One way to attack the problem is by applying statistical language models, which allow us to find sequences of the tokens/words. In previous work, we reached an F-measure of 0.75, identifying only Q1 (anonymized). In this work, we explore the implementation of deep learning architectures to improve the results. For instance, deep learning has had better performance than classic machine learning techniques in the automatic evaluation of student essays [4]. Deep learning is defined as a class of machine-learning algorithms, which uses many layers for representations and transformations of information [5]. Each layer feeds on the output of the previous layer and the hierarchical representations can be obtained by different levels of abstraction. The main contributions of this work are as follows:

- A new annotated corpus with 500 objectives in Spanish of four academic levels: TSU (advanced college-level technician), bachelor, master, and PhD.
- An analysis of sequence labeling with four deep learning architectures: BERT-BETO, CS-ELMo, Multitask Neural Network and Bi-LSTM.
- A detection study for Q1, Q2 and Q3 in a corpus of non-annotated objectives, with the two architectures with best effectivity.

The article is organized as follows: In Section 2, we discuss the work related to our research. The collection used for experimentation is detailed in Section 3. Section 4 includes the methodology followed in the research, while results are presented in Section 5. The detection of the segments corresponding to the different questions is reported in Section 6. The paper is concluded in Section 7, which also includes further work.

2. Related Work

Question Answering (QA) is a task that seeks to identify answers. The first works focused on the creation of templates and predefined rules to identify factual answers. Under an information-retrieval approach, the search is carried out by identifying the entities of interest (topics, named entities) and then, with the extracted features and score functions, producing a ranking [6]. QA has been recently approached with deep learning techniques such as Long Short-Term Memory (LSTM), Recurrent Neural Networks (RNN) [7] and BERT models [8], obtaining good results.

The Complex QA variant performs a deep analysis of the text. For instance, the question "What are the differences between... two approaches" requires a comparison involving the analysis of relationships between components of the text. Two approaches have been identified to address Complex QA: semantic parsing-based methods and information-retrieval-based methods [9]. In our task, the answers respond to open questions established in Q1, Q2 and Q3, although we do not intend to analyze the answer components. However, we have to face the complication that the student can write freely on the subject. Therefore, for example, the answer to Q3 can refer to activities or instruments.

In this work, we address the answer selection problem as a sequence-labeling task using the Inside-Outside-Beginning (IOB/BIO) format. The task of sequence labeling in short texts has been extensively studied in the field of natural language processing, for example, Named Entity Recognition for entity linking, relation extraction, and co-reference resolution. Below, we briefly review previous research that has used the identification of labels in sequences to recognize named entities in short texts. The latter are similar to short answer segments to open questions in objectives, as analyzed in the present paper.

One task that uses short text is the identification of argumentative segments such as premises. In [10], researchers created a corpus of 204 texts annotated with premises, where the texts were collected from social networks. In a first step, they represent each sentence using structural, lexical, contextual, and grammatical features to select only the sentence with argumentation. In a second step, they perform a sequence labeling in each sentence to detect the argumentative fragments. They represent each word using features such as the word, gazetteer lists, and information of verbs and adjectives of argumentative sentences. They achieve a 0.4237 of F1-Measure using Conditional Random Fields (CRF). Also using CRF, researchers of [11] classify segments of text that correspond to argumentative components from news and blogs in the Greek language. They use distributional representations (word2vec), a list of keywords, and part of speech (POS) tags to represent the texts. They report an F-measure of 0.3221 using Conditional Random Fields (CRF). Similarly, we aim to identify text segments in short texts in our work. However, the content of the text segments corresponds to answers to open questions, which are written in free format.

Deep-learning architectures have been also employed for the sequence-labeling task. Algorithms such as CRF, SVM and Bi-LTSM (bi-directional Long Short-Term Memory) are applied in [12], using structural, syntactic, pragmatic, and semantic features. A combination of all features and an architecture that uses several Bi-LTSM networks produces the best results, as they achieve 0.885 of F1 in the detection of argumentative chunks in essays.

Another approach [13] for the identification of argumentative segments in academic essays is the use of deep learning architectures. These are employed to generate contextualized word representations, and are subsequently fed to a CRF network. Traditionally, the CRF model is applied on features manually selected. The reported experiments use a Bi-LSTM-CRF architecture with varied representations such as: Global Vector (GloVe) [14],

Flair [15], ELMo, and BERT, as well as their combinations. The authors achieved an F1-measure of 0.9013 using a Bi-LSTM-CRF with GloVe+Flair+BERT. This demonstrates an improvement when using representations using deep learning architectures. In our proposed solution for answer identification to Q1, Q2 and Q3, we adapted a variant of ELMo that involves an attention mechanism. Furthermore, due to the nature of our corpus, we applied BERT models specific for Spanish.

3. Data Collection

For our experiments, we downloaded the corpus from the Coltypi collection [16]. The corpus includes two kinds of student documents, at graduate-level (master and doctoral) and undergraduate (technical and bachelor). The collected documents that integrate the Coltypi http://coltypi.org/ (accessed on 15 February 2021) collection correspond to theses and research proposals written by students in Spanish, and amount to 968 items. An academic committee has reviewed every item in the collection at some point.

To experiment, we developed a tagging task of 500 objectives. For this task, two annotators were selected with experience in reviewing Spanish academic texts. These annotators teach courses in the area of computing and have been reviewers of student reports and theses. We developed a series of steps for the annotators, considering the recommendations of authors of research methodology books to structure an objective. Moreover, we provide a video using the annotation tool indicating how to mark the text in each objective. Each annotator marked the text segment corresponding to the answers to the methodological questions (Q1, Q2 and Q3). If the annotator does not identify any of them, the Not-Found label was assigned. The Kappa Cohen level of agreement was computed for each methodological question. For Q1, 0.739 was obtained, Q2: 0.696, and Q3: 0.703 with a P level-value less than 0.0001. The strength of agreement reached for the three questions was substantial, where Q1 was that which obtained the highest score. Identifying the text segment of Q1 was probably straightforward for the annotators because this is the action expressed in the objective by the student. The construction of the training and evaluation corpus involves two aspects:

- The first was that the marked objective had the three text segments corresponding to Q1, Q2 and Q3.
- The second aspect was that the text segments matched above 90% of the marked text by annotators.

Finally, out of the 500 tagged objectives, 300 were assigned to the training set and 60 for the test group. A total of 40% of the collection is of graduate-level, and 60% corresponds to undergraduate level. We trained the model with the objectives (360) that fit the three text segments for Q1, Q2 and Q3, simultaneously. In addition, we prepared 597 objectives to carry out a diagnostic experiment with the architectures that performed best. The corpus for the diagnosis was not annotated and was collected exclusively for the detection process, and 46% of the items are of undergraduate level, and the rest (54%) are of graduate level. The annotation process was performed using a proprietary tool. In Figure 1, we present an example of the tool.

The annotator searched for text segments that meet the requirements for Q1, Q2 and Q3. When the annotator marks the text, such a text segment is automatically placed in the boxes to the right of the tool. In the event of an error, the annotator could delete the text by selecting it. In Figure 1, we can observe that the annotator has selected "create a monograph"/"Crear una monografía" and the text appears on the right side in red. The selected text corresponds to the answer to Q1. For each objective, a record is saved in the database using the document number where the objective was extracted and its academic level. Occasionally, some objectives do not have any answer to Q1, Q2 or Q3.

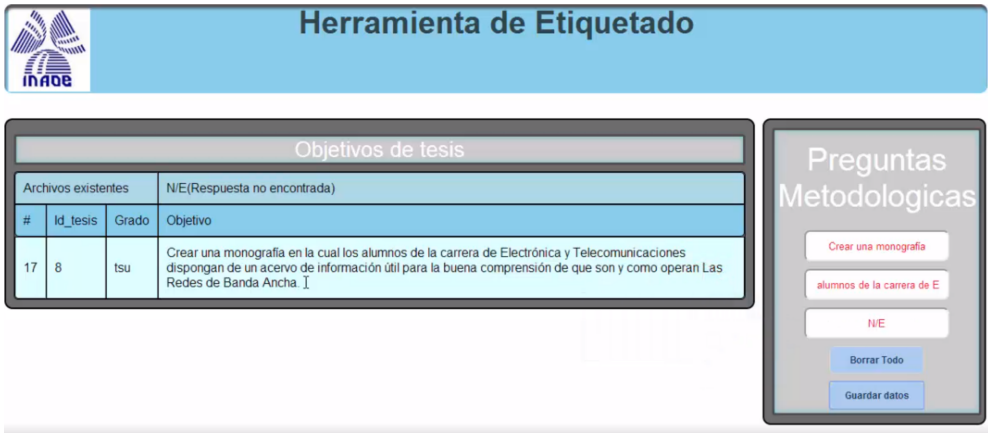

Figure 1. Objective Tagger Tool.

4. Methodology

The collected corpus was used to train deep learning models. Below, in Figure 2, the experimentation scheme is depicted in three components: BIO Tagging, Architectures and Detection.

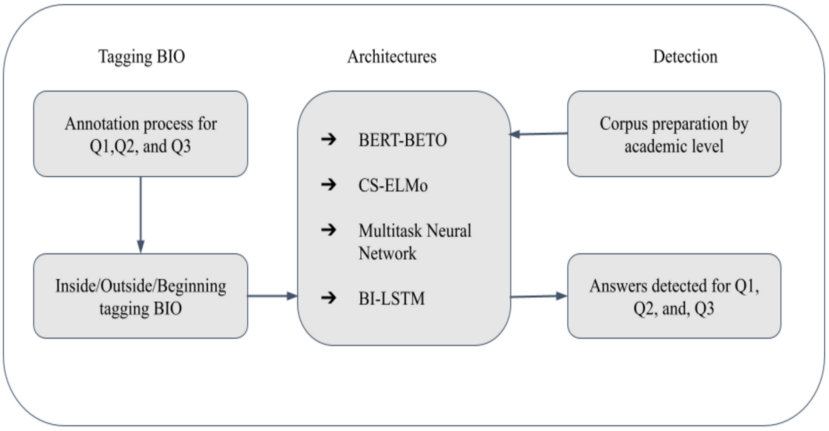

Figure 2. Experimentation scheme.

4.1. Bio Tagging

The first step with the labeled objectives was a tokenization process to separate the text into small units. Then Inside-Outside-Beginning IOB/BIO format was used, which indicates the token that corresponds to the beginning of the methodological response (B), the label (I) that denotes the tokens within the text segment and the label (O) that indicates the tokens that are outside the annotated text segment. Below we show examples for the three questions. We can notice that Q2 has more words marked than the other questions. On average, Q1 contains 8 words, Q2 includes 12 words and Q3 has 9 words.

BIO annotation example for Q1, Q2 and Q3

Q1: [B-Q1]Design [I-Q1]an [I-Q1]architecture [O]that [O]allows [O]the [O]control [O]and [O]monitoring [O]of [O]various [O]robots [O]simultaneously [O]through [O]cloud [O]and [O]serverless [O]tools [O].

Q2: [O]Design [O]an [O]architecture [B-Q2]that [I-Q2]allows [I-Q2]the [I-Q2]control [I-Q2]and [I-Q2]monitoring [I-Q2]of [I-Q2]various [I-Q2]robots [I-Q2]simultaneously [O]through [O]cloud [O]and [O]serverless [O]tools [O].

Q3: [O]Design [O]an [O]architecture [O]that [O]allows [O]the [O]control [O]and [O]monitoring [O]of [O]various [O]robots [O]simultaneously [B-Q3]through [I-Q3]cloud [I-Q3]and [I-Q3]serverless [I-Q3]tools [O].

4.2. Architectures

BERT-BETO: BERT (Bidirectional Encoder Representations from Transformers) is a bidirectional approach that seeks to understand the context of the words. The words to the left and right of each term are employed to train the model. The BERT model [17] has a transformer neural network that infers contextual representations of the words in a text. The BETO model [18] is trained using the BERT paradigm and is similar in size to bert-based-multilingual-cased. The difference is that BETO was trained on the Spanish language. We used BETO-uncased which has 12 self attention layers, with 12 attention heads each. This was trained on Spanish Wikipedia and the sources of the OPUS Project that had text in Spanish. These text sources include the United Nations and government journals, TED talks, subtitles, and news stories. The performance comparison of BETO versus BERT multilingual for Spanish led to favorable results for BETO [18].

CS-ELMo: The second architecture, CSELMo, is an extension of ELMo (Embeddings from Language Models) with an attention mechanism [19]. ELMo is a character-based language model that supplies deep contextualized word representations. ELMo was selected for three reasons. First, it has been trained on many English data sets, and it also extracts morphological information out of character sequences, which is critical for this application since particular character n-grams can identify whether a word belongs to one language. Finally, ELMo develops strong word representations that account for multiple meanings depending on the context. Nonetheless, some characteristics of the standard ELMo architecture could be enhanced to consider additional linguistic properties. Figure 3A describes the overall model architecture, and Figure 3B details the components of the enhanced character n-gram part.

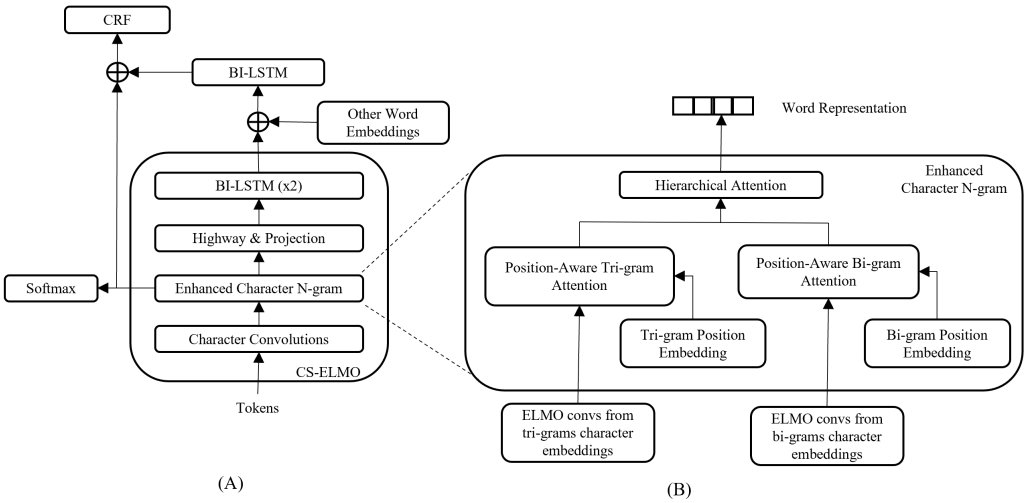

Figure 3. (**A**) The left figure illustrates the overall model architecture. The oversized box describes the components of CS-ELMo. (**B**) The right figure depicts the enhanced character n-gram mechanism inside CS-ELMo [19].

Multitask Neural Network (MTNN): The third architecture MTNN is an approach for Named Entity Recognition, which incorporates in its final layer a CRF box [20]. MTNN can be explained by its three components: feature representation, model description, and sequential inference. The first component, feature representation, aims to select those features standing for the task's most suitable aspects of the data. The features are categorized into characters (an orthographic encoder is used to encapsulate some orthographic features), words (with word embedding models and part-of-speech tags) and lexicons. For the model description, a convolutional neural network architecture is used to learn word shapes and orthographic features. Finally, the last component incorporates the sequential information by using a Conditional Random Fields (CRF) classifier over those probabilities by jointly predicting the most likely sequence of labels for a given sentence.

Bi-LSTM: The last architecture is a Bi-LSTM (bi-directional Long Short-Term Memory), a Recurrent Neural Network [21]. This is based on LSTM (Long Short-Term Memory), which can learn long-term dependencies [22]. Memory cells are used to store information in the LSTM to find and exploit long-distance contexts. LSTM has the capability to add or remove information from the cell state, regulated by gates such as: input, forget and output. A drawback of the LSTM is that it can only utilize the previous context. So in the case of text where we have the previous and following context, we can use both sequences of information. This variation of the LSTM is referred to as bidirectional LSTM (Bi-LSTM). Bi-LSTM takes into account both past and future inputs that are known at a given time, allowing it to act in both right and left directions. In the Bi-LSTM, pre-trained word embeddings, from a Skip-gram model using word2vec, are used for initialization. Additionally, this uses parts of speech (POS) tag embeddings that are initialized randomly according a uniform distribution. The Bi-LSTM is fed with the concatenation of both the word embeddings and POS tags. As mentioned, the Bi-LSTM considers the features of both directions backward and forward and concatenates the resulting vectors of each direction.

4.3. Detection

For this purpose, we have two processes, the first is the preparation of a different corpus to that employed for training and evaluation. The preprocessing was performed in the same way as indicated in Section 4.1, splitting the objectives by academic level (i.e., graduate and undergraduate).

In the second process, after evaluating the different architectures, we selected two of the architectures with the best efficacy (BERT-BETO and CS-ELMo). The detection was carried out to have a diagnosis of the percentage of segments found in Q1, Q2 and Q3 and to have a comparison between the academic levels. We also compare two of the architectures in a corpus not considered for training.

4.4. Hyperparameter Configuration

The final parameters configured with the applied architectures are detailed in the following paragraphs.

BERT-BETO: Training was carried out under the 10-fold cross-validation method. The parameters with the best performance were: 160 epochs, 3×10^{-5} of learning rate, and a batch size of 16. However, the number of epochs during the fine-tuning was 20, 80, 160 and 200; the batch size was calculated with 16 and 32. In particular, we employed the model described in [18].

Bi-LSTM: Dropout layers with a dropout of 0.1 are added before and after the Bi-LSTM layers. A softmax activation was configured with size units of 100. For embedding, we selected a dimension of 50. The training was carried out also under the 10-fold cross-validation scheme.

MTNN: We train the entire neural network with a batch of 300 samples and 300 epochs using the AdaMax optimizer. On both convolutional layers, we use kernel size of 3. On both layers, we apply 64 filters in each layer. The Bi-LSTM architecture included dropout layers before and after the Bi-LSTM layers using a dropout rate of 0.5. The CRF classifier uses L-BFGS as a training algorithm, employing an L1/L2 regularization. Penalties of 1×10^{-3} for L2 and 1.0 for L1 are applied.

CS-ELMo: A batch size of 10 samples and 30 epochs is used to train the entire model using the Adam optimizer. We select n-gram order of 3 and frozen ELMo for fine tuning.

F1-score, precision and recall were computed as evaluation metrics to assess the efficacy of the different architectures. Precision is the percentage of segments found by the learning system that are correct. Using this metric, we can measure the positive patterns that have been correctly predicted from the total predictions [23]; it is defined as follows:

$$Precision = \frac{tp}{tp + fp}$$

Recall is the percentage of segments present in the corpus that are found by the system. This measure is the fraction of positive patterns that are correctly classified [23], defined as follows:

$$Recall = \frac{tp}{tp + tn}$$

Finally, F1-score consists of the harmonic mean between recall and precision values, expressed as:

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

5. Experimentation Results and Discussion

The results obtained in each of the proposed architectures and for each question are shown below. In each experiment carried out, the precision, recall, and F-measure metrics were obtained to conduct a comparison. The results obtained were the best with each architecture applied.

5.1. Results for Q1

We can observe in Table 1 that the CS-ELMo architecture obtains the best F-measure results to identify the beginning of the B-Q1 segment and for I-Q1. In the precision columns, the MTNN architecture obtains the best result for B-Q1, and for I-Q1, BERT-BETO works better. In the recall columns, the CS-ELMo architecture obtains the best results for B-Q1 and I-Q1. We identify an acceptable consistency of results to select CS-ELMo as the architecture that performs best in identifying Q1. Finally, we can observe similar results of the BERT-BETO architecture compared with CS-ELMo.

Table 1. Results for question Q1: What will you do?

Architectures	Precision			Recall			F-Measure		
	B-Q1	**I-Q1**	**O**	**B-Q1**	**I-Q1**	**O**	**B-Q1**	**I-Q1**	**O**
BERT-BETO	0.968	**0.895**	0.953	0.956	0.800	**0.980**	0.962	0.848	0.966
CS-ELMo	0.967	0.840	**0.991**	**1.000**	**0.949**	0.969	**0.984**	**0.891**	**0.980**
MTNN	**0.980**	0.640	0.970	0.950	0.810	0.920	0.970	0.720	0.940
Bi-LSTM	0.680	0.210	0.980	0.290	0.680	0.900	0.400	0.320	0.940

5.2. Results for Q2

We can see in Table 2 that the BERT-BETO architecture obtains the best F-measure and precision results to identify the beginning of the segment B-Q2 and for I-Q2. In the recall columns, the Bi-LSTM architecture obtains the best results for B-Q2, and BERT-BETO for I-Q2 segments. So, we identify a consistent efficacy of BERT-BETO architecture, which is selected as the best for identifying the answers to Q2.

Table 2. Results for question Q2: What is the purpose of doing it?

Architectures	Precision			Recall			F-Measure		
	B-Q2	**I-Q2**	**O**	**B-Q2**	**I-Q2**	**O**	**B-Q2**	**I-Q2**	**O**
BERT-BETO	**0.733**	**0.754**	0.919	0.797	**0.833**	0.876	**0.761**	**0.789**	0.897
CS-ELMo	0.593	0.674	0.829	0.633	0.615	0.859	0.612	0.643	0.844
MTNN	0.680	0.610	0.810	0.530	0.600	0.820	0.600	0.600	0.820
Bi-LSTM	0.280	0.470	**0.950**	**0.850**	0.390	**0.950**	0.420	0.420	**0.950**

5.3. Results for Q3

We can observe in Table 3 that the BERT-BETO architecture obtains the best F-measure results to identify the start of the segment (B-Q3) and for I-Q3. In the precision columns, the MTNN architecture obtains the best result for B-Q3, but BETO-BERT performs better for the I-Q3 segment. In the recall columns, the Bi-LSTM architecture obtains the best results

for B-Q3, and BERT-BETO for the I-Q3 segment. As expressed by the F-measure columns, we identify good consistency in selecting BERT-BETO as the architecture that performs best in identifying the answers to Q3.

Table 3. Results for question Q3: How are you going to do it?

Architectures	Precision			Recall			F-Measure		
	B-Q3	I-Q3	O	B-Q3	I-Q3	O	B-Q3	I-Q3	O
BERT-BETO	0.794	**0.816**	0.935	0.848	**0.771**	0.783	**0.821**	**0.783**	0.943
CS-ELMo	0.793	0.746	0.940	0.766	0.759	0.938	0.779	0.752	0.939
MTNN	**0.800**	0.630	0.900	0.620	0.600	0.920	0.700	0.610	0.610
Bi-LSTM	0.120	0.340	**0.970**	**0.850**	0.190	**0.950**	0.220	0.250	**0.960**

Analyzing the results of the best architectures, we find that BERT-BETO obtains most of them in Q1, Q2 and Q3, followed by CS-ELMo for F-measure. These two architectures have the characteristic of considering positional information. On one hand, CS-ELMo has an enhanced character n-gram module which incorporates position embeddings, position-aware attention, and hierarchical attention. On the other hand, BERT in the transformer model uses input embeddings which include the position of the word in the sequence: the first word/token, the size of the word. A property we observed in BERT is that it uses a Bidirectional Encoder Representations from Transformers, which helps identify when a word changes meaning as the sentence unfolds. This feature of BERT could mainly help in the proper identification of I-Q2 and I-Q3 tags. However, we also identify that the MTNN architecture performs well in identifying Q1 and Q3; the start of the text segment corresponds to one word/token. It is worth mentioning that the CS-ELMo architecture uses a multilingual dataset, while BERT-BETO was pre-trained with a corpus specific to Spanish.

6. Detection of Segments Q1, Q2 and Q3

Once we obtained the best models for the identification of segments corresponding to Q1, Q2 and Q3, we opted to develop the detection experiment using the methods BERT-BETO and CS-ELMo. We performed a diagnosis on a corpus of non-annotated objectives. As described in Section 3, there were 597 objectives collected from different levels. For the TSU level, 88 were collected, the undergraduate level 189, the master's level 256, and the doctorate level 64 objectives. In Table 4, we can observe the results for the detection of Q1, i.e., segments that answer the question "What is going to be done". We noticed that the BERT-BETO architecture detected more B-Q1 labels in three of the four academic levels. For the I-Q1 segment, BERT-BETO obtained a larger number of segments detected in the four levels.

Table 4. Detection results for question Q1: What is going to be done?

Architectures	CS-ELMo			BERT-BETO		
	B-Q1	I-Q1	O	B-Q1	I-Q1	O
TSU	85	770	5493	111	838	5159
Bachelor	219	1875	14,259	213	2037	13,577
Master	268	2040	12,169	294	2169	11,608
PhD	64	627	5031	91	702	4755

We performed a qualitative analysis on the predictions and found that BERT-BETO had a more accurate detection of the B-Q1 tag than CS-ELMo. In Table 5, we present an example in which CS-ELMO places the label B-QUE right at the beginning of the objective in the word (uno) "One" while BERT-BETO places it in the infinitive verb (crear) "create". According to institutional guidelines for preparing a thesis, an objective must begin with a verb in the infinitive form. We notice that BERT-BETO begins the segment where the objective should start, leaving aside the text that does not correspond. BERT-BETO has

higher precision in the majority of the predictions. We observed that BERT-BETO did not label any B-QUE tag in objectives without verbs; on the other hand, CS-ELMo wrongly assigned the label B-QUE.

Table 5. Segmentation example for B-Q1 and I-Q1.

Example of bachelor objective: One of the most important objective of the project was to create a set of hardware with independent routines that would configure themselves at runtime in a way that is transparent to programming. (Spanish to English translation)			
CS-ELMo segmentation		BERT-BETO segmentation	
One	B-QUE	One	O
of	I-QUE	of	O
the	I-QUE	the	O
most	I-QUE	most	O
important	I-QUE	important	O
objective	I-QUE	objective	O
of	I-QUE	of	O
the	I-QUE	the	O
project	O	project	O
was	O	was	O
to	O	to	O
create	I-QUE	create	B-QUE
a	I-QUE	a	I-QUE
set	I-QUE	set	I-QUE
of	I-QUE	of	I-QUE
hardware	I-QUE	hardware	I-QUE
.....	.....	.....	.....

In Table 6, we observed that BERT-BETO obtained the largest number of identified segments for the B-Q2 label, which indicates the beginning of the text segment for Q2. The same behavior is shown for the I-Q2 segment. It is worth mentioning that the words labeled as O are those that complement the sentence of the objective. In some cases, they are connectors. After comparing both architectures, we notice by looking at the predictions that both architectures labeled the same segments. In other words, the two architectures have a good efficacy in the task.

Table 6. Detection results for question Q2: Why is it going to be done?

Architectures	**CS-ELMo**			**BERT-BETO**		
	B-Q1	**I-Q1**	**O**	**B-Q1**	**I-Q1**	**O**
TSU	119	1830	4399	131	1872	4290
Bachelor	216	3713	12,464	238	3370	12,703
Master	245	3661	10,571	251	4497	9645
PhD	71	1109	4548	82	1479	4139

Finally, we observe the results for Q3 in Table 7. We notice that for label B-Q3, CS-ELMo obtains a slightly higher number of identified segments when compared with BERT-BETO. An important point is that within each objective some students wrote more than two objectives; that is the reason why columns B-Q1, B-Q2 and B-Q3 have more labels than the number of objectives in each level. However, this behavior can be fixed by the human reviewer.

Table 7. Detection results for question Q3: How is it going to be done?

Architectures	CS-ELMo			BERT-BETO		
	B-Q1	I-Q1	O	B-Q1	I-Q1	O
TSU	85	1337	4926	104	1367	4838
Bachelor	145	3295	12,953	141	2837	13,313
Master	161	2639	11,677	170	2318	11,949
PhD	65	1579	4084	57	987	4653

Figure 4 on the Y-axis shows the number of segments detected by the architectures with the best performance. On the X-axis, the academic levels are specified. The segments detected by academic level are depicted on each of the circles. We include the Begin (B) and In (I) labels of Q1, Q2 and Q3 in the sum. The objective of the graph is to visualize which architecture has more coverage to identify Q1, Q2 and Q3. In a light gray color, we can observe that BERT-BETO is slightly above CS-ELMo for the bachelor level. For TSU, master and PhD BERT-BETO obtains a better efficacy. However, as mentioned above, after reviewing the predictions, the BERT-BETO model performs a more accurate detection.

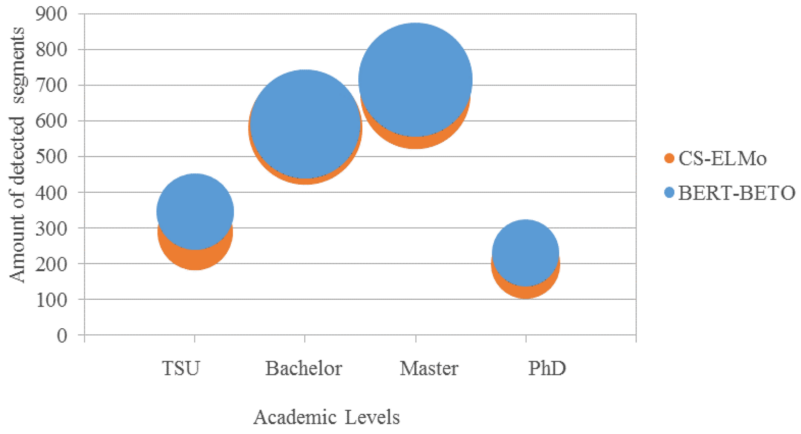

Figure 4. Amount of segments identified by the CS-ELMo and BERT-BETO architectures.

In addition, we compute the percentage of segments detected for Q1, Q2 and Q3 between academic levels. We obtained the average for the results of two compared architectures. We found that bachelor, master and PhD levels reached 100% for question Q1 (What is going to be done?), and for TSU it reached 95%. It should be noted that when models identified two Q1 labels, we only counted once. In the case of TSU level, we find objectives where Q1 was not detected.

For question Q2 (Why is going to be done?), we found that the master level reached a 96% detection rate. TSU, bachelor and PhD levels reached 100%. For question Q3, we found that only the PhD level reached 100%; that is, each analyzed objective contained at least one text segment that answered the question "How is it going to be done?". The master's level obtained 62%, the bachelor's level obtained 76%, and the TSU level obtained a level of 95%.

After reviewing the results, we noticed that segments Q1 and Q2 were detected in all four academic levels. However, we observe that for Q3, only the doctorate level reaches 100%. In Q3, it is expected that the method, technique or tool to develop the project or thesis will be defined. We also observed a good result for the objectives evaluated at the TSU level. At the technical level, students are expected to finish in a period of two years and have mostly practical training; this approach may allow students to identify the tools to use in their project.

7. Conclusions

Here, we proposed and tested a new variant of question answering beyond factual and definition questions, specifically for the kind of methodological questions detailed. This, to our knowledge, has not been done before.

Evaluating objectives is a task that requires knowledge of a specific topic and experience on the part of the academic reviewer. In our work, we focused on analyzing the objectives of student theses in the area of computing, finding that deep-learning architectures such as BERT-BETO and CSELMo have an acceptable performance in terms of capturing the structure of the answers to the three open questions that were tackled in this work. We note that for question Q1, the F-score results were the highest, while Q2 and Q3 were close to 0.8 of F1. It is worth mentioning that answers for Q2 and Q3 are larger than Q1 and have more variety in terms of vocabulary; therefore, it is more complex to capture a single pattern. An alternative to improve the results would be to increase the number of examples for questions Q2 and Q3.

An expected result in detecting the non-annotated corpus was that the PhD level reached a high percentage of detection in the answers to the three questions, possibly due to a more demanding academic training. However, an unexpected result was that the TSU level obtained more consistent Q1, Q2 and Q3 detection values than the bachelor and master levels. One difference observed was that PhD-level objectives were longer than TSU-level objectives. We believe that the TSU student program, which is two years of training and is practice-oriented, allows students to identify the answers to Q2 and Q3 in less time. A relevant detail is that this training is oriented to problem solving in offices or companies.

The methodology developed in this work could be addressed to analyze objectives in other types of texts closer to theses and research proposals, such as technical articles in the same domain. We plan to test our trained architectures on a set of papers. Moreover, the methodology can also be applied in other areas far from computing, after feeding a corpus with new instances.

In a future scenario, we foresee a pilot test with the BERT-BETO and CS-ELMo architecture, thus taking advantage of the best efficacy in each open question. We want to include students of different academic levels from graduate and undergraduate levels. It will be interesting to observe the behavior of these architectures in an actual environment, both in terms of performance and feedback. The results obtained are promising, so our method can be a complementary tool for academic reviewers; thus, the review time of a thesis could be shorter, and the reviewer would focus on aspects with deep content.

Author Contributions: Conceptualization, S.G.-L., Z.G.M.-R., A.P.L.-M., A.L.-L. and J.M.G.-G.; Data curation, S.G.-L., Z.G.M.-R. and J.M.G.-G.; Formal analysis, S.G.-L. and A.L.-L.; Funding acquisition, S.G.-L., A.L.-L. and J.M.G.-G.; Investigation, S.G.-L., Z.G.M.-R., A.P.L.-M., A.L.-L. and J.M.G.-G.; Methodology, Z.G.M.-R.; Project administration, A.P.L.-M.; Resources, J.M.G.-G.; Supervision, S.G.-L., A.L.-L. and J.M.G.-G.; Validation, A.P.L.-M.; Writing—original draft, S.G.-L., Z.G.M.-R. and J.M.G.-G.; Writing—review & editing, A.P.L.-M. and A.L.-L. All authors have read and agreed to the published version of the manuscript.

Funding: This research has not received any external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available upon reasonable request from the corresponding author.

Acknowledgments: We want to thank the annotators of the collection. All authors was partially supported by SNI-Conacyt.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. López, S.G.; López-López, A. Mining Domain Knowledge for Coherence Assessment of Students Proposal Drafts. In *Educational Data Mining*; Springer: Berlin/Heidelberg, Germany, 2013; pp. 229–255.
2. González-López, S.; López-López, A. Analysis of Concept Sequencing in Student Drafts. In *Open Learning and Teaching in Educational Communities*; Springer: Berlin/Heidelberg, Germany, 2014; pp. 422–427.
3. Calijorne Soares, M.A.; Parreiras, F.S. A literature review on question answering techniques, paradigms and systems. *J. King Saud Univ. Comput. Inf. Sci.* **2020**, *32*, 635–646. [CrossRef]
4. Shin, J.; Gierl, M.J. More efficient processes for creating automated essay scoring frameworks: A demonstration of two algorithms. *Lang. Test.* **2021**, *38*, 247–272. [CrossRef]
5. Deng, L.; Yu, D. Deep Learning: Methods and Applications. *Found. Trends Signal Process.* **2014**, *7*, 197–387. [CrossRef]
6. Fu, B.; Qiu, Y.; Tang, C.; Li, Y.; Yu, H.; Sun, J. A Survey on Complex Question Answering over Knowledge Base: Recent Advances and Challenges. CoRR; 2020. Available online: http://xxx.lanl.gov/abs/2007.13069 (accessed on 10 April 2021).
7. Ishwari, K.S.D.; Aneeze, A.K.R.R.; Sudheesan, S.; Karunaratne, H.J.D.A.; Nugaliyadde, A.; Mallawarachchi, Y. Advances in Natural Language Question Answering: A Review. CoRR; 2019. Available online: http://xxx.lanl.gov/abs/1904.05276 (accessed on 5 March 2021).
8. Wang, C.; Luo, X. A Legal Question Answering System Based on BERT. In Proceedings of the 2021 5th International Conference on Computer Science and Artificial Intelligence, Beijing, China, 4–6 December 2021; Association for Computing Machinery: New York, NY, USA, 2021; pp. 278–283.
9. Lan, Y.; He, G.; Jiang, J.; Jiang, J.; Zhao, W.X.; Wen, J. A Survey on Complex Knowledge Base Question Answering: Methods, Challenges and Solutions. CoRR; 2021. Available online: http://xxx.lanl.gov/abs/2105.11644 (accessed on 7 May 2021).
10. Goudas, T.; Louizos, C.; Petasis, G.; Karkaletsis, V. Argument extraction from news, blogs, and social media. In Proceedings of the Hellenic Conference on Artificial Intelligence; Springer: Berlin/Heidelberg, Germany, 2014; pp. 287–299.
11. Sardianos, C.; Katakis, I.M.; Petasis, G.; Karkaletsis, V. Argument Extraction from News. In Proceedings of the 2nd Workshop on Argumentation Mining, Denver, CO, USA, 4 June 2015; ACL: Seattle, WA, USA, 2015; pp. 56–66.
12. Ajjour, Y.; Chen, W.F.; Kiesel, J.; Wachsmuth, H.; Stein, B. Unit Segmentation of Argumentative Texts. In Proceedings of the 4th Workshop on Argument Mining, Copenhagen, Denmark, 8 September 2017; ACL: Seattle, WA, USA, 2017; pp. 118–128.
13. Petasis, G. Segmentation of argumentative texts with contextualised word representations. In Proceedings of the 6th Workshop on Argument Mining, Florence, Italy, 1 August 2019; ACL: Seattle, WA, USA, 2019; pp. 1–10.
14. Pennington, J.; Socher, R.; Manning, C.D. Glove: Global vectors for word representation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014; ACL: Seattle, WA, USA, 2014; pp. 1532–1543.
15. Akbik, A.; Blythe, D.; Vollgraf, R. Contextual string embeddings for sequence labeling. In Proceedings of the 27th International Conference on Computational Linguistic, Santa Fe, NM, USA, 20–26 August 2018; pp. 1638–1649.
16. González-López, S.; López-López, A. Colección de Tesis y Propuesta de Investigación en TICs: Un recurso para su análisis y estudio. In Proceedings of the XIII Congreso Nacional de Investigación Educativa, Chihuahua, Mexico, 16–20 November 2015 2015; pp. 1–15.
17. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Minneapolis, MN, USA, 2–7 June 2019; Association for Computational Linguistics: Minneapolis, MN, USA, 2019; Volume 1, pp. 4171–4186.
18. Cañete, J.; Chaperon, G.; Fuentes, R.; Ho, J.H.; Kang, H.; Pérez, J. Spanish Pre-Trained BERT Model and Evaluation Data. In Proceedings of the PML4DC at ICLR 2020, Addis Ababa, Ethiopia, 26–30 May 2020.
19. Aguilar, G.; Solorio, T. From English to Code-Switching: Transfer Learning with Strong Morphological Clues. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online, 5–10 July 2020; pp. 8033–8044.
20. Aguilar, G.; Maharjan, S.; López-Monroy, A.P.; Solorio, T. A Multi-task Approach for Named Entity Recognition in Social Media Data. In Proceedings of the 3rd Workshop on Noisy User-generated Text, Copenhagen, Denmark, 7–9 September 2017; Association for Computational Linguistics: Copenhagen, Denmark, 2017; pp. 148–153.
21. Zhang, S.; Zheng, D.; Hu, X.; Yang, M. Bidirectional Long Short-Term Memory networks for relation classification. In Proceedings of the 29th Pacific Asia Conference on Language, Information and Computation, Shanghai, China, 30 October–1 November 2015; pp. 73–78.
22. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef] [PubMed]
23. Goutte, C.; Gaussier, E. A Probabilistic Interpretation of Precision, Recall and F-Score, with Implication for Evaluation. In Proceedings of the Advances in Information Retrieval, Santiago de Compostela, Spain, 21–23 March 2005; Losada, D.E., Fernández-Luna, J.M., Eds.; Springer: Berlin/Heidelberg, Germany, 2005; pp. 345–359.