ISSN 2007-9737

Rule-Based Spanish Multiple Question Reformulation and their Classification Using a Convolutional Neural Network

Alberto Iturbe Herrera, Noé Alejandro Castro Sánchez, Dante Mújica Vargas

Tecnológico Nacional de México,
Centro Nacional de Investigación y Desarrollo Tecnológico,
Mexico

iturbe@cenidet.edu.mx, {noe.cs, dante.mv}@cenidet.tecnm.mx

Abstract. Question reformulation allows the creation of different forms of the same question in order to identify the best answer. However, when aspects such as length and complexity increase, the reformulation process becomes more complicated, consequently also the recovery of the corresponding information. In this research, a method for the reformulation of multiple questions in Spanish is presented, as part of the pre-processing stage in a question-answer system. The lexical category of each word, Named Entities and Multi-Word Terms, were used to reformulate multiple questions into new individual questions, and then a Convolutional Neural Network was used to classify them, allowing to find or build adequate answers to improve the quality of the results, which is fundamental in QA systems. A dataset with multiple questions was also created to evaluate our reformulation method, since it was not possible to find any. On the other hand, for the evaluation of the question classification model, we used the TREC, Simple Questions, Web Questions, Wiki Movies and Curated TREC datasets, translated into Spanish. Both tasks achieved promising results for further work.

Keywords. Question reformulation, question classification, convolutional neural networks.

1 Introduction

Due to the accelerated growth of information available in digital format, the company Seagate in conjunction with the consultancy IDC, in 2018 published the report The Digitalization of the World From Edge to Core [22], in which it is estimated that by 2025 there will be five times more data than in 2018, an approximate of 175 ZB, of which 79% will be in text format. Nowadays, search mechanisms have made progress that allows the extraction of direct answers to simple queries. However, when aspects such as complexity or length of the queries come into play, the probability of obtaining the desired information on the first web page or document retrieved is less and less.

This is where the Question-Answering (QA) task is introduced, the objective of which is to provide users with the best answer from a given query, that is, to generate concise answers, unlike the search mechanisms that retrieve a set of web pages or related documents. However, the aforementioned problems also wreak havoc on QA Systems. Above all, if the input questions are complex in terms of the number of questions in these (multiple questions), taking into account that they need to be classified or identify important sections as the focus that the question, that is, the statement or phrase that makes the question flow.

The reformulation and classification of questions have been extensively researched. Seeking to automate these tasks, different approaches have been proposed. On the part of the reformulation of questions, the use of patterns stands out as in [29] which introduced a model capable of providing concrete answers by implementing reformulation techniques based on semantics to retrieve the corresponding answer from the huge number of documents retrieved by an engine of search. In [33] they presented a method for the automatic extraction of reformulation patterns from *5w1h* questions, in addition evaluating the

search performance of this type of questions using previously generated patterns. On the other hand, [21] proposed a model capable of improving the performance of a QAS through the semantic reformulation of questions. This work focused on the identification of standard reformulation patterns to find the exact candidate answer.

Recently, in [2] a QA system called question answering environment model was introduced, which is capable of reformulating the question written by the user and subsequently obtaining the correct answer. In [15] the Answer-Supervised Question Reformulation model was presented, which improves automatic conversational understanding by implementing reinforcement learning using *pointer-copy* as a question reformulation model. Finally, in [11] a framework called MSReNet was proposed for the open-domain question answering task.

This framework improved the performance of the QA task using a multi-step reformulator that generates a new question using the available passages and question. Similarly, on the question classification side, there are works that used the pattern recognition approach, such as [19], who presented a semantic kernel based on knowledge that uses WordNet as a knowledge base and a measure for semantic relation (SR).

On the other hand, with the machine learning or deep learning paradigm, there are works such as [18] which presented an analysis of questions based on their grammatical structure. In [10], experiments over a Graph-based approach to the question answering task based on entrance exams on CLEF 2014 was presented, obtaining a c@1 of 0.375.

In this work, different patterns were identified using machine learning algorithms to classify them. In [31] the pipeline of the AOQA system for list and factoid questions was analyzed, emphasizing the classification of questions with multiple labels. At [12] a question classification model for the Chinese language was presented. This model combines the bi-attention mechanism and LSTM networks.

[17] presented a new deep neural network model called Attention-Based BiGRU-CNN (ABBC) for the classification of Chinese-language questions. This model combines the characteristics and advantages of the convolutional neural network, the attention mechanism and the recurrent neural network, which allows extracting the characteristics of the questions in Chinese. Finally, in cite Xu2020 they introduce a larger challenge dataset for the QC task, with 7787 science test questions labeled based on a hierarchical taxonomy.

The work above solves the tasks involved with different techniques achieving good results. However, most of these are focused on the English language, except for two research studies for the Chinese language.

On the other hand, none of the cited works solves the problem of multiple questions, a type of question that encapsulates in itself two or more questions such as: *¿quién fue y cuándo nació el fundador de* $<$*ENTITY*$>$*?*((Who was and when and where was the founders of ENTITY born?)) This makes tasks such as the identification of elements, classification of questions and the retrieval of answers difficult, especially in languages other than English which does not have the same amount of information.

Motivated by the above, we present a method for the reformulation of multiple questions using the lexical category of each word, Named Entities and Multi-Word Terms. This method is performed in the pre-processing stage of a question-answering system with the aim of reformulating a multiple question into new individual questions according to the elements identified throughout the sentence.

Subsequently, a Convolutional Neural Network is evaluated for the classification of questions using the existing TREC, Simple Questions, Web Questions, Wiki Movies and Curated TREC datasets. In addition, the new research questions are generated.

This document is structured as follows: Section 2 shows the solution method for reformulating and classifying questions. The experiments and results are described in Section 3. And finally, Section 4 describes the conclusions and future work of this research.

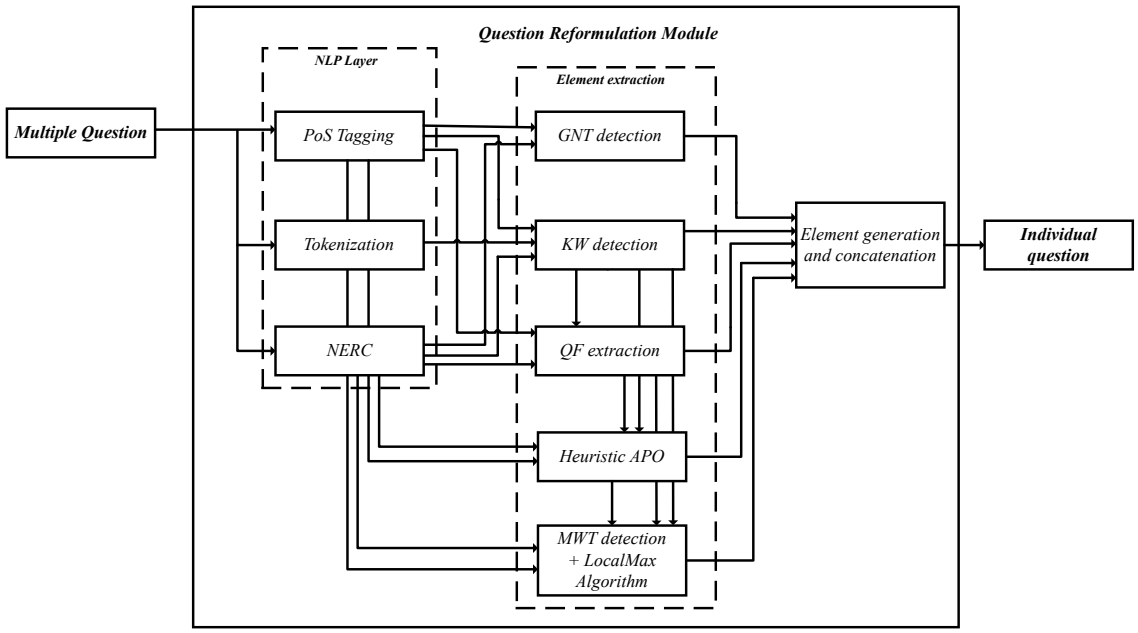


Fig. 1. Module for question reformulation

2 Solution Method

Within the Question-Answering task, neural models have achieved excellent results, however, most of these are focused on solving the problem for the English language. However, regardless of the language, the length and complexity of the sentences are factors that have a direct impact on the acquisition of good results in most of the NLP tasks, including the QA, as demonstrated in [6, 14].

Similarly, regardless of technology, search engines are also affected by the variables mentioned above. This was observed when querying two or more questions to these engines. An example of these types of questions is:

¿Quiénes fueron y cuándo y dónde nacieron los fundadores de <ENTITY>?

Who were and when and where were the founders of ENTITY born?; these are called complex questions in terms of the number of questions present in the sentence, that is, multiple questions, *plurium interrogationum* [30]. These types of questions, unlike those of the factoid type which can be answered with simple facts expressed in short text answers, require more complex procedures such as the decomposition of the question and the summary of multiple documents to retrieve the corresponding answers [5].

As a result of the above, search engines tend not to extract the correct answers for all questions formulated in the same sentence, that is, the greater the complexity of the question, the lower the probability that the correct answer is drawn.

Let us consider the examples: *¿Quién fue conocido como el padre de la computación y cuándo murió?* (Who was known as the father of computers and when did he die?), *¿Cuál fue, cuándo y dónde se creó el primer lenguaje de programación?* (Which, when and where was the first programming language created?) or *¿Cuáles son los lenguajes de programación más utilizados para minería de datos y por qué?* (What are the most widely used programming languages for data mining and why?).

These are multiple questions, where the complexity of each one is different due to the presence of elements such as: multi-word terms, named entities, question focus, sentence length, among others.

Motivated by this, a model was developed to identify and classify the elements of complex questions and to reformulate them into individual questions. The proposed method is divided into 5 parts as shown in Figure 1: construction of a corpus of multiple questions using Wikipedia; a question pre-processing module; module of reformulation of questions that includes the extraction and classification of elements, as well as the generation and concatenation of new elements; and finally, the question classification module using a Convolutional Neural Network.

2.1 Creation of the Corpus

Wikipedia in Spanish gathers a total of 1,636,082 articles, annexes and categories, ranking it in ninth place in Wikipedias based on the total content pages. Figure 2 shows the amount of articles in the Spanish Wikipedia over the years.

This makes Spanish Wikipedia a great source of information, which allows us to construct a corpus using this information, limiting it to the domain of computer science and related areas.

WikiExtractor[1] was used to extract and clean the information from the database of that source. WikiExtractor converts Wikipedia articles into plain text files or json files, which makes it possible to have a set of files free of HTML tags.

This tool is invoked using a *Wikipedia dump* file[2]. The output is stored in different files, which will contain various articles in xml and json format.

Additionally, it was identified that some Wikipedia articles contain a section called *infobox*[3], here the content of the site is summarized allowing the reader to find fragments of information more easily. Taking into account the tabulated nature of this section, questions were generated from the elements it contains. So the main topic is identified by being the heading of the table; the items in the left column are Question Focus; and those in the right column are the immediate answers.

[1] https://github.com/attardi/wikiextractor

[2] These files can be downloaded from the following link: https://dumps.wikimedia.org/

[3] Example of Wikipedia article with Infobox: https://es.wikipedia.org/wiki/Python

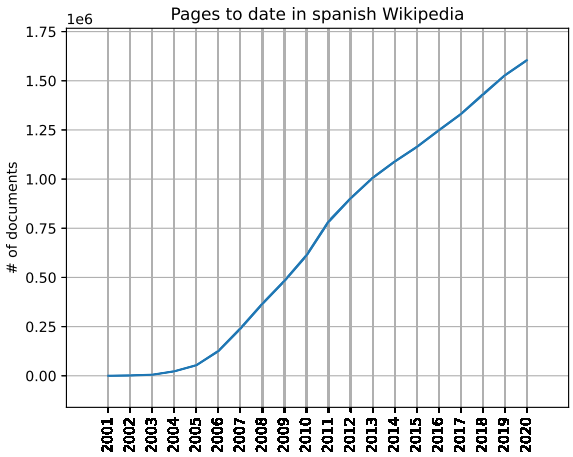


Fig. 2. Spanish Wikipedia statistics

Table 1. Total questions per class

ENT	NUM	LOC	DESC	HUM
131	70	84	76	139

The generation process implements the text analysis described in 2.2 and the algorithm described in 2.3.

With this information, a corpus of 500 questions was generated which were classified according to the question classes of the TREC classification dataset. Table 1 shows the distribution of the questions in these classes which corresponds to Entity, Numeric, Location, Description and Abbreviation, respectively.

It is important to note that the 500 questions generated are simple questions, that is, they have a single interrogative term. Multiple questions like: *¿Cuándo apareció Python y cuáles son las extensiones comunes?* (When did Python appear and what are the common extensions?), *¿Cuál es la última versión estable de Python y por quién fue influido?* (What is the latest stable version of Python and by whom was it influenced?) or *¿Cuándo apareció Python y cuáles son las extensiones comunes?* (When did Python appear and what are the common extensions?) were generated manually by combining two or more elements of the corpus.

Therefore, to create a multiple question, the selection of between two and three questions on the same topic is used as a starting point. Of the two questions shown in Figure 3 the interrogative words are identified and positioned in such a way that all the questions maintain coherence among themselves, being separated by commas and the conjunction *y*. Followed by each interrogative word, the focus of the question is embedded in case the focus of each question is different (3a). On the contrary, if the focus of the question matches with the others, it is added after the last question word (3b). Finally, the main topic or entity on which the questions are being asked can be added immediately after the first question focus (3a) or at the end of the sentence (3b).

Taking this into account, two resulting corpus were obtained: the first of these is composed of simple questions, which was used to evaluate the classification of questions in Spanish using a Convolutional Neural Network; and the second corpus, which consists of 1140 multiple questions, was used to evaluate the module of reformulation of complex/multiple questions.

2.2 Analysis and Treatment of Questions

The objective of this stage is to apply Natural Language Processing techniques to identify the elements present in a multiple question in order to reformulate it into individual questions for classification with a Convolutional Neural Network. This stage receives as input a multiple question from the corpus described in the previous section. This process begins with the extraction of the grammatical category of each word, as well as the parallel execution of the tokenization and the Named Entity Recognition and Classification using *Freeling* [20].

The tasks already mentioned provide fundamental information for each of the different grammatical categories. For example, Table 2 shows the PoS tag assigned to each word by Freeling based on the categories proposed by *EAGLES*[4] for tagging. The information obtained allows the module to identify the different PoS tags and thus be able to reformulate the multiple questions into individual elements. As an example, the tag ***VSIS3S0*** provides the following information: (***V***) determines the category, in this case verb; The next (***S***) represents the type, semi-auxiliary; the third position (***I***) corresponds to the mood, imperative; later the tense is indicated(***S***), past; immediately the person is specified (***3***), third; followed by this, if plural(***P***) or singular(***S***); finally, the gender (***0***) [5].

On the other hand, for the NERC, in addition to Freeling, a heuristic of item skip parsing was developed, improving the the Named Entities recognition. This heuristic was also used for the generation of the individual questions.

2.3 Question Reformulation

Once the above information was obtained, an algorithm was developed to classify the grammatical elements in the sentence, as well as for the generation of adjectives, determiners, pronouns or adverbs, identifying the gender, number and tense from the PoS labels obtained, this allows to create a correctly structured sentence.

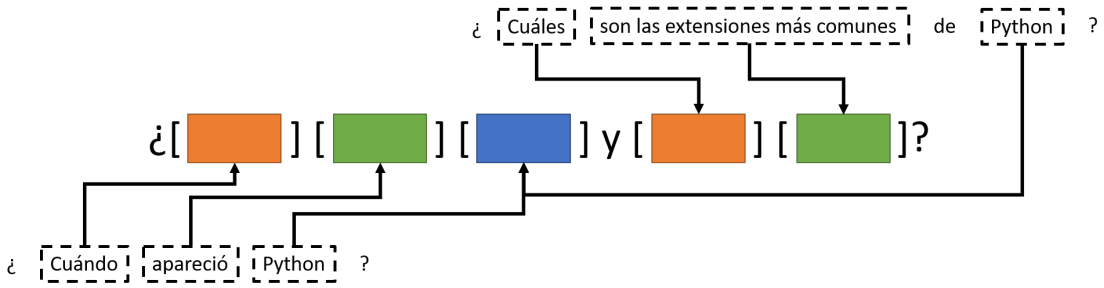
In addition, the algorithm performs a search for keywords throughout the sentence, these, in most cases, correspond to the intention of the question based on the interrogative term. This allows identifying in the first instance the possible number of questions in the current sentence.

In addition, considering that a certain type of question can use different keywords, lists of words were created according to the type of question, for example: fecha[*cuándo, fecha, año, mes, día*], ubicación[*dónde, lugar, país, sede*], among others.

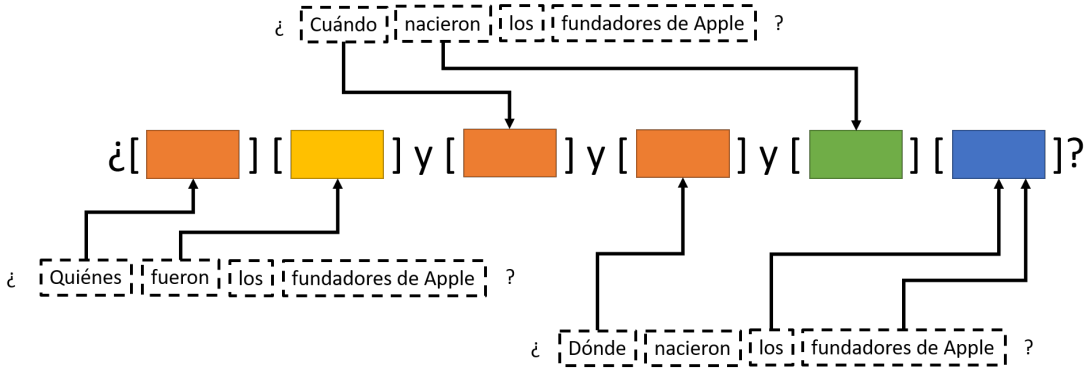
Once the previous information was obtained, it was necessary to locate the Focus of the Question, which is a statement or phrase that makes the question flow. This element must be related to the content or the desired results. For this, the algorithm LocalMax [8] was used on the corpus built. LocalMax algorithm assumes that Multi-Word Terms (MWT) have a high rate of adherence to each other.

[4] https://www.cs.upc.edu/ nlp/tools/parole-sp.html

[5] More information about PoS tags can be found at: *https://www.sketchengine.eu/spanish-freeling-part-of-speech-tagset/*



(a) Two questions



(b) Three questions

Fig. 3. Multiple question construction

The authors propose an association metric called Symmetrical Conditional Probability (SCP) to measure the correlation between two words using Equation 1:

$$SCP = p(x|y) \cdot p(y|x) = \frac{p(x,y)^2}{p(x) \cdot p(y)}, \qquad (1)$$

where $p(x,y)$; $p(x)$; and $p(y)$ are the probabilities that the bigram (x,y), the unigram $p(x)$ and the unigram $p(y)$ appear in the corpus respectively.

To generalize this measure of n-grams, the authors introduce the Fair Dispersion Normalization, that splits an n-gram $w_1, w_2, ..., w_n$ at different points of dispersion and considers it as combinations of the two parts.

FDS uses the average of the products to normalize the association measure for a given n-gram (Equation 2). To measure the cohesion between words in an n-gram, the average of the products for the two parts at different points of dispersion of the n-gram is calculated (Equation 3):

$$SCP\_f\left(w_1 w_2 \dots w_n\right) = \frac{p\left(w_1 w_2 \dots w_n\right)^2}{Avp}, \qquad (2)$$

$$Avp = \frac{1}{n-1} \sum_{i=1}^{n-1} p\left(w_1 \dots w_i\right) \cdot p\left(w_{i+1} \dots w_n\right), \qquad (3)$$

where n is the length of the n-gram and $p(w_1, \dots, w_n)$ is the probability of the sequence of words w_i.

Based on FDS, the LocalMax algorithm tries to find an n-gram with the highest SCP_f that any (n-1)-gram in this and any (n + 1)-gram containing it, allowing to identify MWT and with that avoid the loss of information. To this process the heuristic of item skip parsing is added to this process using stop words, PoS and NERC tags.

Once classified, a set of patterns were constructed for the reformulation of questions. This

Table 2. PoS tags

Token	Tag	Description
¿	Fia	*pos=punctuation, type=questionmark, punctenclose=open*
Qué	PT00000	*pos=pronoun, type=interrogative*
es	VSIP3S0	*pos=verb, type=semiauxiliary, mood=indicative, tense=present, person=3, num=singular*
Java	NP00SP0	*pos=noun, type=proper, neclass=person*
y	CC	*pos=conjunction, type=coordinating*
cuándo	PT00000	*pos=pronoun, type=interrogative*
fue	VSIS3S0	*pos=verb, type=semiauxiliary, mood=indicative, tense=past, person=3, num=singular*
creado	VMP00SM	*pos=verb, type=main, mood=participle, num=singular, gen=masculine*
?	Fit	*pos=punctuation, type=questionmark, punctenclose=close*

is done by adding new items and concatenating existing ones to give sense to individual questions. In most cases, grammatical elements such as articles, prepositions or determiners are added. These are necessary to generate a coherent sentence, as required. Table 3 shows 8 out of a total of 19 generated patterns.

Table 3. Example patterns for question reformulation

Question patterns
['FIA', 'PT', 'VS', 'DA', 'AO','QF, MWT, NE', 'FIT']
['FIA', 'PT', 'VS', 'DA', 'NP','QF, MWT, NE', 'FIT']
['FIA', 'PT', 'P0', 'VM', 'DA','QF, MWT, NE', 'FIT']
['FIA', 'SP', 'PT', 'VS','QF, MWT, NE', 'FIT']
['FIA', 'SP', 'PT', 'VS', 'VM','QF, MWT, NE', 'FIT']
['FIA', 'SP', 'PT', 'VM', 'DA', 'QF, MWT, NE', 'FIT']
['FIA', 'DT', 'QF, MWT, NE', 'FIT']
['FIA', 'DI', 'QF, MWT, NE', 'FIT']

In Table 3, *FIA* represents the opening question mark (¿), *PT* corresponds to the Interrogative Word that determines the type of question, *VS,VM, P0, DT,. . . ,n* indicate the verb, determinant or information to generate a coherent sentence. In the case of infoboxes: *QF, NE* y *MWT* Question Focus, Named Entities and Multi-Word Term respectively, are determined based on the position in the table, the left column (QF) and the infobox title (NE); finally, *FIT* represents the closing question mark (?). The importance of this reformulation lies in the individual analysis of words to identify enough information to generate coherent questions, respecting aspects of gender and number agreement, as well as the proper use of verbs, determiners, prepositions, among others. Additionally, this allows us to eliminate words that are not strictly necessary in individual questions.

2.4 Word Embedding Creation

Taking into account the need to use word vectors in Spanish, it was decided to use Spanish Billion Word Corpus [4], which contains around 1.5 billion words collected from different information sources such as: SenSem, Ancora Corpus, Tibidabo Treebank and IULA Spanish Treebank; the OPUS project comprising: the books aligned by Andras Farkas, the compilation of legislative texts of the European Union by the CCI, the News Commentary corpus and United Nations documents compiled by Alexandre Rafalovitch and Robert Dale; the Spanish portion of the European Parliament by Philipp Koehn; and finally, snippets from Wikipedia, Wikisource and Wikibooks up to 2015, using Wikipedia Extractor.

Therefore, the process for its creation was replicated using FastText [3], which is a library written in C++ for the efficient learning of word representations and sentence classification. FastText allows you to train supervised and unsupervised representations of words and sentences. These can be used for different data compression applications as features in additional

models, for candidate selection or as initializers for transfer learning [9].

In this work, the Skip-gram model was used using negative sampling and softmax. This tool achieves good performance for the aforementioned tasks, but also in the case of rare words using information at the character level.

2.5 Question Classification Method

The importance of identifying the type of question that will be processed by a QAS is fundamental for this task, as shown [7, 27, 28, 16, 19, 18, 31, 12, 17, 32]. Consequently, a Convolutional Neural Network model was used to classify questions in Spanish.

To understand how a CNNs work in [1, 23, 23] this architecture is described, in which each layer of the network is three-dimensional, having a spatial extent and depth corresponding to the number of features. The notion of depth of a single layer in a CNN is different from the notion of depth depending on the number of layers.

In the input layer, these features correspond to color channels like RGB (red, green, blue), and in the hidden channels these features represent hidden feature maps that encode various types of shapes in the image. The architecture contains two types of layers: convolution and pooling.

For convolution layers, a convolution operation must be defined. Afterward, a filter to assign triggers from one layer to the next is used. A convolution operation uses a three-dimensional weight filter with the same depth as the current layer, but with a smaller spatial extent. The dot product between all the weights in the filter and any spatial region options (the same size as the filter) in one layer defines the value of the hidden state in the next layer (after applying a trigger function like ReLU). The operation between the filter and the spatial regions of a layer is performed at every possible position to define the next layer (in which the triggers retain their spatial relationships from the previous layer).

Generally, in the Natural Language Processing task for text classification [13] uses filters that move over entire rows of a word-based matrix. Where, the width of each filter is usually the same as the width of the input array. The height or size of the region can vary, but the sliding of the windows is usually two to five words at a time.

In this research, different sets of questions of p words and the length of the input sentences defined by n words were used, where each word is represented by a vector, in this case a Word Embedding in Spanish language described in 2.4.

Therefore, an array of $n \cdot k$ is obtained, where k corresponds to the length of the Word Embedding (300). To this, the batch is added and it is denoted by b, which means the total number of inputs (questions) that the neural network will process simultaneously. A padding is applied to all those sentences that do not match a previously defined maximum length. For the convolution operation, the aforementioned input $n \cdot k$ is used, now called x, and a matrix of weights $W(m \cdot k)$, which generates as output a vector h that is obtained with the Equations 4 and 5:

$$h_{i,1} = \sum_{j=1}^{m} \sum_{l=1}^{k} w_{j,l} x_{i+j-1,l}, \tag{4}$$

$$h = W * x + b. \tag{5}$$

In turn, different filter sizes are defined that are executed in parallel. Each convolution outputs a hidden vector of dimension $1 \cdot n$, which is concatenated to generate the input of the next layer with a dimension $q \cdot n$, where q is the number of parallel layers that will be used. Therefore, assuming the output of the layer h is of dimension $q \cdot n$ the pooling layer will produce an output of size $q \cdot 1$ denoted as h' (Equation 6):

$$h'_{i,1} = \left\{ \max \left(h^{(i)} \right) where 1 \leq i \leq q \right\}. \tag{6}$$

Table 4. Distribution of questions by class

	Classes					
	ENT	**DESC**	**PERS**	**NUM**	**LOC**	**ABR**
TREC	1,052	1,042	1,033	823	735	77
WM	45,441	33,459	11,700	5,263	319	3
WQ	542	494	374	202	154	0
SQ	37,012	18,983	18,648	10,752	1,272	1
CTREC	381	327	296	246	212	24

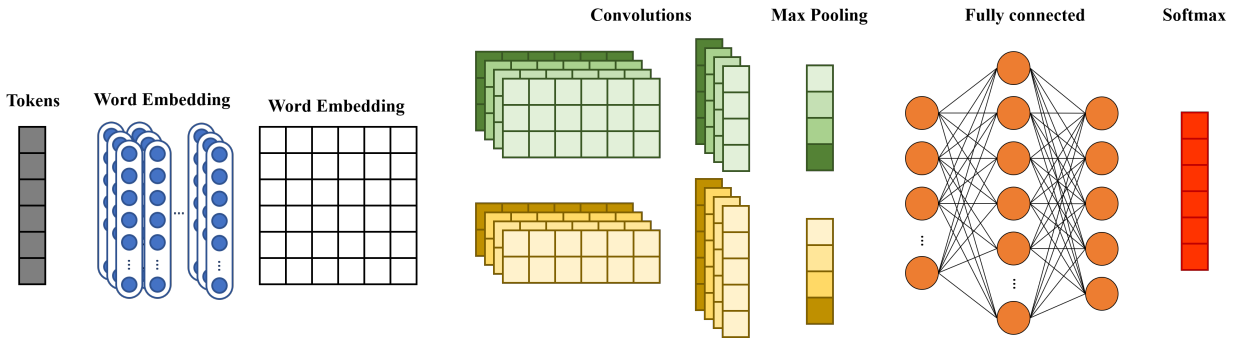


Fig. 4. CNN architecture

Table 5. Average score for each BLEU category

BLEU-1	BLEU-2	BLEU-3	BLEU-4
0.8839	0.8371	0.7976	0.6113

Figure 4 shows the architecture used for this Convolutional Neural Network. This shows the vector corresponding to the tokens of the input sentence, followed by the Word Embedding used to calculate the vector of real value for each word using its 300 dimensions.

Subsequently, the convolution and max pooling layers to finally reach the fully connected and softmax layer. With a total of 100 filters with dimensions of sizes 2, 3 and 4, an embedding layer dimension defined at 300, with an input size defined based on the vocabulary size, a dropout of 0.5, applying 2D convolutions. In this way, the neural network will be able to classify the Spanish questions.

Of the aforementioned datasets, only TREC focuses on the question classification task, therefore, the classes defined for this dataset were: Entity, Description, Person, Numeric, Location and Abbreviation.

The rest of the datasets were classified manually and with the help of an algorithm based on the use of keywords for each category. Table 4 shows the total elements per class for each dataset.

3 Experiments and Results

This section describes the experiments performed to evaluate the proposed method. These are divided into two parts: the first shows the results for the reformulation of multiple questions using the metrics BLEU, ROUGE, METEOR and WER; the second part shows the results of the classification of the previously reformulated questions using a CNN in a QA system.

3.1 Question Reformulation Evaluation

To evaluate the reformulation of questions, 250 questions in Spanish were randomly selected from the corpus of multiple questions generated in 2.1. To measure the effectiveness of the method, the following metrics were selected based on the literature. Although these were originally created to evaluate other tasks of the NLP, they have achieved good results in this task. These metrics perform the calculation considering a reference sentence and a candidate sentence.

3.1.1 BLEU

Also known as BiLingual Evaluation Understudy (equation 7) It was originally designed to evaluate the quality of translations made by computer systems. Table 5 shows the average scores obtained for each BLEU category, where: 1, 2, 3 and 4 represent the sequence of n-grams to

evaluate an input, unigrams, bigrams, trigrams, among others:

$$BLEU = PB \cdot \exp\left(\sum_{n=1}^{N} w_n \log P_n\right). \quad (7)$$

It is important to mention that, if the number of words in the input is less than the number of n-grams to be evaluated, the BLEU value is automatically zero.

3.1.2 METEOR

Metric for Evaluation of Translation with Explicit ORdering like BLEU was originally designed to evaluate automatic translation. This metric is based on the harmonic mean of precision and recovery of unigrams reaching an average score of 0.8805. Table 6 shows some examples of the score obtained between a candidate sentence against the reference sentence.

3.1.3 ROUGE

Recall-Oriented Understudy for Gisting Evaluation commonly known as ROUGE (Equations 8-10) was originally designed for the evaluation of automatic translation and automatic summarization tasks.

For this research, ROUGE-N (1 and 2) and ROUGE-L were used, which can be considered as the granularity of the texts that are compared between the output of the system and the reference elements. For example, ROUGE-1 and ROUGE-2 refer to the overlapping of unigrams and bigrams, respectively, between the system output and the reference output. Unlike ROUGE-L, which measures the longest word sequence using Longest Common Subsequence (LCS):

$$precision = \frac{number\_of\_overlapping\_words}{total\_words\_in\_system\_output}, \quad (8)$$

$$recall = \frac{number\_of\_overlapping\_words}{total\_words\_in\_reference\_input}, \quad (9)$$

$$F_1 = 2 \cdot \frac{precision \cdot recall}{precision + recall}. \quad (10)$$

Table 7 shows the average precision, recall and F-Measure obtained by each ROUGE category.

3.1.4 Word Error Rate (WER)

Finally, Word Error Rate (better known as WER) is a metric originally designed to evaluate the performance of a speech recognition or automatic translation systems. WER (Equation 11) is derived from the Levenshtein distance, working at the word level rather than the phoneme level:

$$WER = \frac{(S + D + I)}{N} = \frac{(S + D + I)}{(S + D + C)}, \quad (11)$$

where *S*, *N*, *I*, *D* y *C* represent the number of substitutions, total reference words, insertions, deletions and correct words in the output, respectively. Some examples of candidate questions and the WER value obtained with respect to the reference question are shown in Table 8.

Table 9 shows the average values of correct words, substitutions, deletions, insertions and the 'WER index in the corpus.

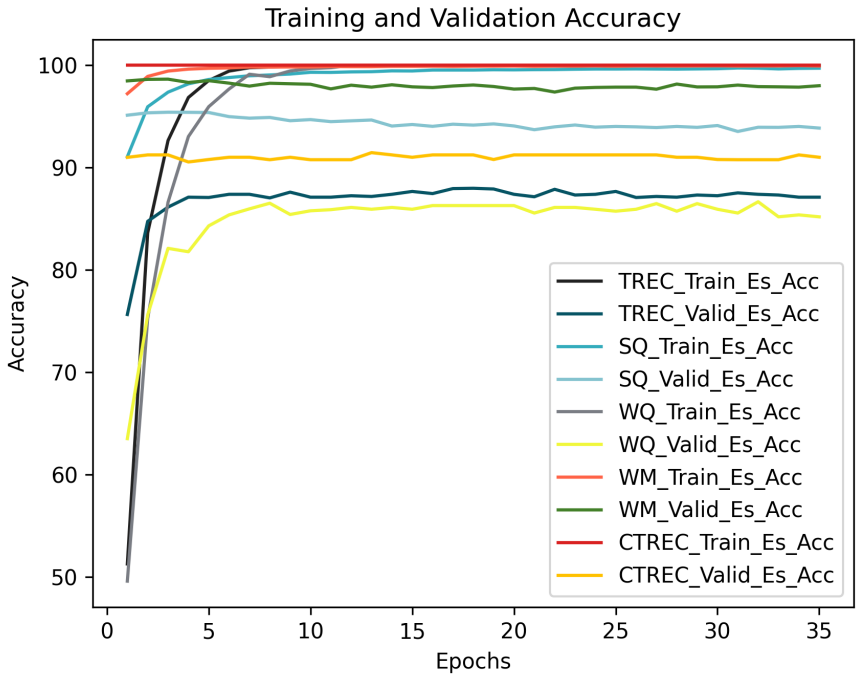
3.2 Question Classification Evaluation

Taking into account the small amount of information in Spanish for training the Convolutional Neural Network, each of the aforementioned datasets was translated with the help of online tools.

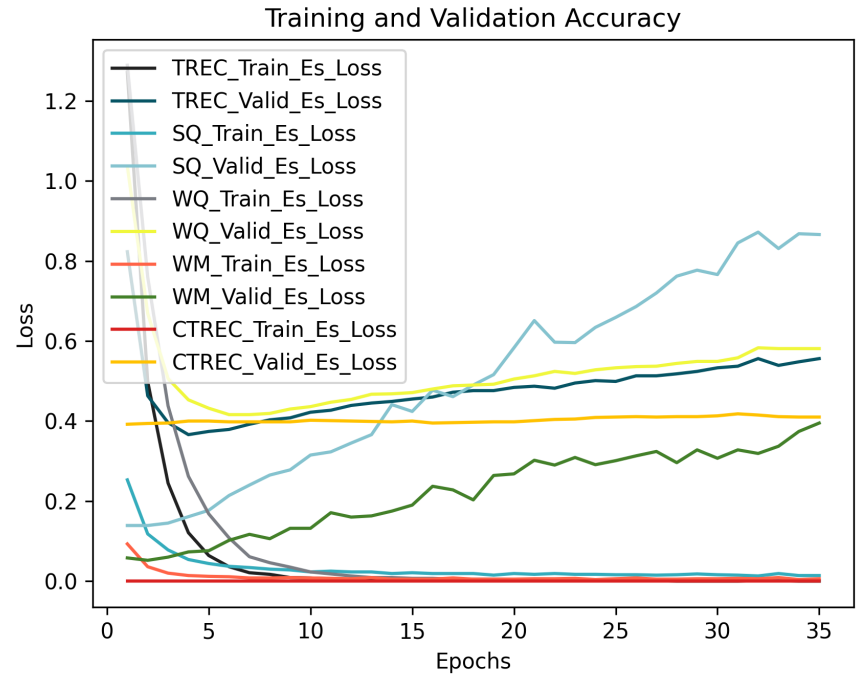
For the evaluation, the CNN was trained individually for each corresponding dataset for 35 epochs. Figure 5 shows the Accuracy and Loss values at each epoch for each dataset in the training and validation sets.

In Figure 5a the corresponding accuracy is shown. Here it can be seen that the set with the highest value was Curated TREC in the training set at 100% throughout the 35 epochs, which means an overfitting in the Neural Networks. On the other hand, the WebQuestions validation set had an accuracy above 80%.

Also in Figure 5b the loss is shown for each training and validation set according to the corresponding dataset in the 35 epochs. Here it is possible to see that the set with the lowest loss was Curated TREC in training, which could support that the model has been overfitted. On the other hand, the SimpleQuestions validation set was the one with the highest loss.



(a) Dataset Accuracy



(b) Dataset Loss

Fig. 5. Accuracy and Loss obtained in training and validation of each dataset used

Table 6. Sample questions and their METEOR scores obtained

Reference	Candidate	METEOR
¿Cuál es la sede de **X** corporation? (What is the headquarters of **X** corporation?)	¿Dónde es la sede de **X** corporation? (Where is the headquarters of **X** corporation?)	0.8552
¿Qué nacionalidad tuvo alan turing? (What nationality did alan turing have?)	¿Qué nacionalidad es alan turing? (What was the nationality of alan turing?)	0.7500
¿Cuáles son las extensiones comunes de r? (What are common extensions of r?)	¿Cuáles son las extensiones comunes de? (What are common extensions of?)	0.7217

Table 7. Average score of Precision, Recall and F-measure for each ROUGE category

	Average score		
	Precision	**Recall**	**F-Measure**
Rouge-1	0.9101	0.8903	0.8985
Rouge-2	0.8569	0.8391	0.8461
Rouge-L	0.9120	0.8911	0.8999

Below, a set of Figures (6-10) are shown corresponding to the confusion matrices for each dataset respectively in the test set, this tool allows to visualize the performance of the implemented CNN. Each column in the matrix represents the percentage of predictions for each class, while each row represents the instances in the actual class. This tool allows us to see if the system is confusing which classes.

Figure 6 shows the corresponding confusion matrix for the Curated TREC dataset. Each class is represented by the corresponding number: {'NUM': 0, 'LOC': 1, 'PER': 2, 'ENT': 3, 'DESC': 4, 'ABR': 5}. Here it is possible to see that classes 0, 1 and 2 are those with the best classification results. However, classes 3, 4 and 5 present a deficiency in the classification, this is due mainly to the syntactic similarities between each type of question. In future, we plan to apply syntactic n-grams for analysis of these similarities [26, 24, 25].

For the SimpleQuestions V2 dataset the following notation was used: {'ENT': 0, 'PER': 1, 'LOC': 2, 'DESC': 3, 'NUM': 4}.

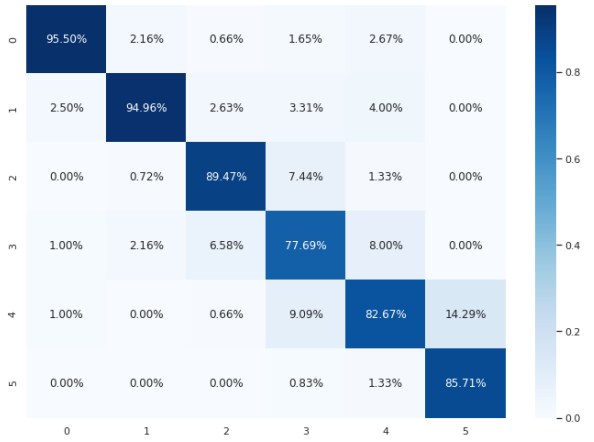


Fig. 6. Curated TREC confusion matrix

Omitting class 5 due to the absence of elements. Unlike the previous figure, Figure 7 shows more uniform results in the classification of classes 0, 1, 2 and 3. However, the classifier presents a significant error in class 4.

The confusion matrix shown in Figure 8 corresponds to the TREC 10 dataset. The following notation was used in it: {'DESC': 0, 'ENT': 1, 'PER': 2, 'NUM': 3, 'LOC': 4, 'ABR': 5}. For this dataset, classes 2, 3, and 4 had a lower error rate. However, class 5 had a clearly questionable performance, this is associated with the low amount of training data.

On the other hand, Figure 9 shows the results for the WebQuestions dataset, and, as with SimpleQuestions, class 5 is ignored for the same reason. Using the notation: {'ENT': 0, 'PER': 1, 'LOC': 2, 'DESC': 3, 'NUM': 4} there is a clear

Table 8. Sample questions and their obtained WER scores

Reference	Candidate	C	S	D	I	WER
¿Cuándo nació el fundador de **N**? (When was the founder of **N** born?)	¿Cuándo nació el fundador de **N**? (When was the founder of **N** born?)	6	0	0	0	0
¿Qué es la minería de datos? (What is data mining?)	¿Qué es minería de datos? (What is data mining?)	5	0	1	0	1
¿Quién es el desarrollador de mariadb? (Who is the developer of mariadb?)	¿Quién es Maria el desarrollador de db? (Who is Maria the db developer?)	5	1	0	1	1

Table 9. Average of words for each WER item

AVG C	AVG S	AVG D	AVG I	AVG WER
9	3	3	2	3

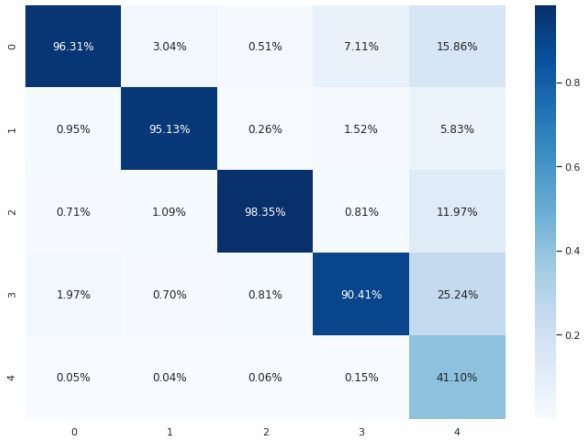


Fig. 7. SimpleQuestions confusion matrix

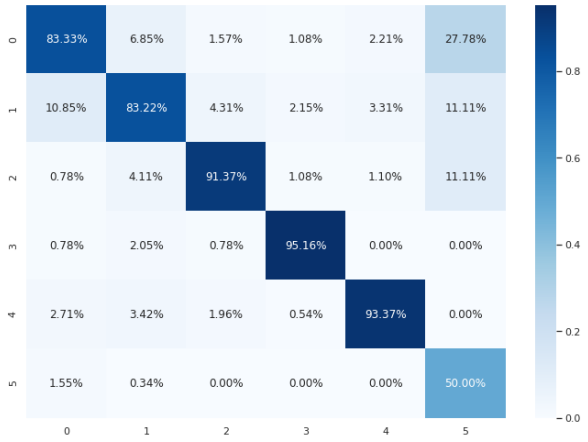


Fig. 8. TREC 10 confusion matrix

balance between classes. However, class 3 shows a higher percentage of error.

Finally, Figure 10 displays the confusion matrix for the WikiMovies dataset with the notation: {'PER': 0, 'ENT': 1, 'NUM': 2, 'DESC': 3, 'LOC': 4}, obtaining good results except for class 4.

Next, Table 10 shows the accuracy and loss values for each dataset respectively, in the validation set. This shows that Curated TREC achieved the highest accuracy percentage with a 98.82%, as well as the lowest loss with 0.044. On the contrary, SimpleQuestions had the lowest accuracy with 87.80% and WebQuestions had the highest loss with 0.347.

Taking into account that the questions reformulated in 2.3 may contain errors that interfere with the identification of the type of question, the manual classification and the prediction of the corresponding class was carried out using CNN. Figure 11 show the corresponding confusion matrix.

Here it is possible to appreciate that classes 2 and 3 are those with less precision in the classification. This is due to the syntactic similarity of both kinds of questions.

3.3 Discussion of the Results

The previously described metrics have shown promising results. For example, both ROUGE

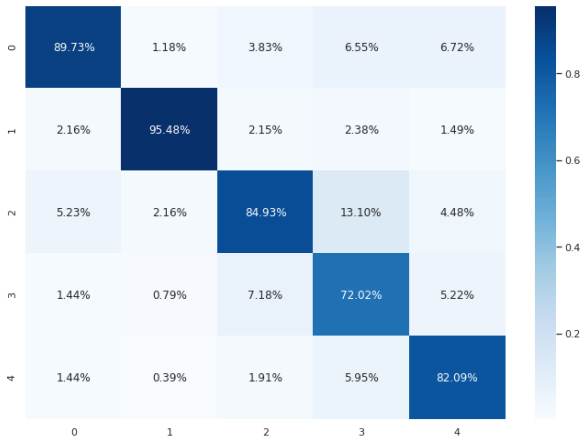


Fig. 9. WebQuestions confusion matrix

and BLEU rely on n-grams to measure the similarity between the output generated by a system (candidate) and the output generated by a human (reference). However, the difference between ROUGE-n and BLEU is that BLEU introduces a penalty term, and also calculates n-gram matching for various sizes of n-grams, as opposed to ROUGE-n, where there is only one size of n-gram.

Furthermore, METEOR has several features not found in other metrics, such as stemming and synonymous matching, along with standard exact word matching. This has reached a good correlation with human judgment at the sentence level. Finally, WER was used in order to identify the range of correct words, substitutions, deletions and the average number of word errors per sentence.

Therefore, taking into account that the average results of these metrics are similar, it is possible to conclude that, in general terms, the reformulation of the questions tends to be closer to the reference sentences, showing good results.

For the question classification it was possible to determine that, indeed, for the uni-class classification, the variables of the length and complexity of the questions have a direct impact on the correct classification.

In the same way, since CNN has not been trained with examples with these characteristics, the multiple questions are not correctly classified. In general, an accuracy of 98.82%, 87.98%, 94.96%, 88.94% and 87.80% was obtained for the Curated TREC, WebQuestions, WikiMovies, TREC 10 and SimpleQuestions datasets respectively.

Table 10. Accuracy and loss in validations sets

	CTREC	WQ	WM	TREC	SQ
Accuracy	98.82%	87.98%	94.96%	88.94%	87.80%
Loss	0.04	0.34	0.14	0.30	0.32

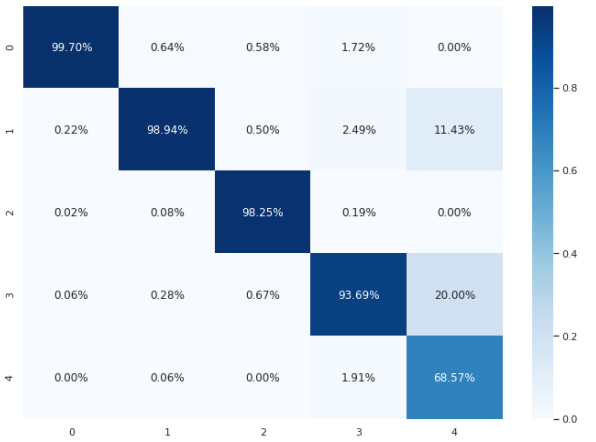


Fig. 10. WikiMovies confusion matrix

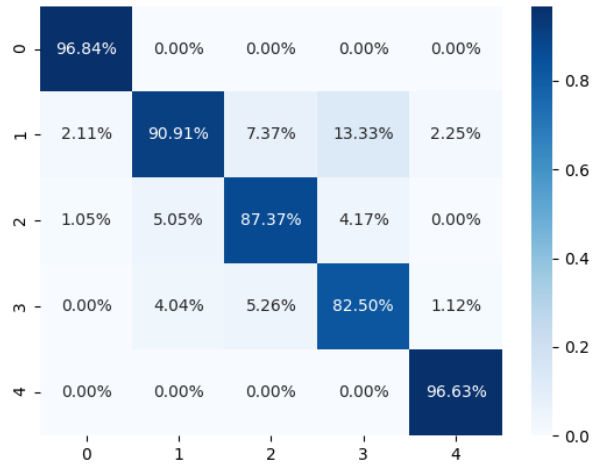


Fig. 11. Confusion matrix of question reformulation

On the other hand, for the questions that were previously reformulated, the results by class were 96.84%, 90.91%, 87.37%, 82.50% and 96.63% respectively.

However, when a multiple question was submitted to CNN, the class in more than 80% of the cases was entity type, whether or not this type of question was present.

This confirms the importance of question reformulation for this research.

4 Conclusions and Future work

This work presented an approach based on Pattern Recognition and Natural Language Processing for question reformulation from the identification and classification of elements in multiple questions as a stage that precedes the classification of questions of a QA system using a CNN.

It is important to mention that the works cited in section 1 carry out the reformulation of questions and their classification to improve the retrieval of answers in the English language, which is why it is not possible to carry out the comparison of results taking into account the properties and difficulties of each language.

Although this approach obtained good results, it could be improved with the implementation of Artificial Neural Network models. Therefore, as future work, a larger corpus, adaptations for other languages, as well as the respective models of Artificial Neural Networks will be built.

In addition, a corpus of text strings with two or more questions was built that could be used both in future works of this research and for other investigations that are carried out in the Processing of Natural Language.

On the other hand, we work on a neural model capable of identifying the *n* classes and providing an adequate classification of questions within a multiple question.

Finally, the question reformulation tends to become complicated in a directly proportional way according to the length of the sentences, the number of questions in them, the Named Entities present, the Focus of the Question and the multi-word terms.

Acknowledgments

This work was partially supported by the government of Mexico (CONACYT grant, SNI).

References

1. **Aggarwal, C. C. (2018).** Neural Networks and Deep Learning - A Textbook. Springer.
2. **Ali, I., Yadav, D. (2019).** Question reformulation based question answering environment model. International Journal of Information Technology.
3. **Bojanowski, P., Grave, E., Joulin, A., Mikolov, T. (2017).** Enriching word vectors with subword information. Transactions of the Association for Computational Linguistics, Vol. 5, pp. 135–146.
4. **Cardellino, C. (2019).** Spanish Billion Words Corpus and Embeddings.
5. **Chali, Y., Hasan, S. (2012).** Simple or complex? classifying questions by answering complexity. Proceedings of the Workshop on Question Answering for Complex Domains, The COLING 2012 Organizing Committee, Mumbai, India, pp. 1–10.
6. **Cho, K., Merriënboer, B., Bahdanau, D., Bengio, Y. (2014).** On the properties of neural machine translation: Encoder–decoder approaches. Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation, Association for Computational Linguistics, pp. 103–111.
7. **Dodiya, T., Jain, S. (2016).** Question classification for medical domain question answering system. 2016 IEEE International WIE Conference on Electrical and Computer Engineering (WIECON-ECE), pp. 204–207.
8. **Ferreira da Silva, J., Dias, G., Guilloré, S., Pereira-Lopes, J. (1999).** Using LocalMaxs algorithm for the extraction of contiguous and non-contiguous multiword lexical units. Progress in Artificial Intelligence, Springer Berlin Heidelberg, pp. 113–132.
9. **Gómez-Adorno, H., Posadas-Durán, J., Sidorov, G., Pinto, D.**, **.** Document embeddings learned on various types of n-grams for cross-topic authorship attribution. Computing, Vol. 100, No. 7, pp. 741–756.

10. **Gómez-Adorno, H., Sidorov, G., Pinto, D., Gelbukh, A. F. (2014).** Graph based approach for the question answering task based on entrance exams. **Cappellato, L., Ferro, N., Halvey, M., Kraaij, W.**, editors, Working Notes for CLEF 2014 Conference, Sheffield, UK, September 15-18, 2014, volume 1180 of CEUR Workshop Proceedings, CEUR-WS.org, pp. 1395–1403.
11. **Han, W., Peng, M., Xie, Q., Zhang, X., Wang, H. (2020).** Msrenet: Multi-step reformulation for open-domain question answering. **Zhu, X., Zhang, M., Hong, Y., He, R.**, editors, Natural Language Processing and Chinese Computing, Springer International Publishing, Cham, pp. 292–304.
12. **Hu, S., Du, Y., Luo, X., Kong, D., Li, Q. (2019).** A hybrid bi-attention mechanisms and long short-term memory model for chinese question classification. 2019 2nd International Conference on Safety Produce Informatization (IICSPI), pp. 609–612.
13. **Kim, Y. (2014).** Convolutional Neural Networks for Sentence Classification. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP).
14. **Koehn, P., Knowles, R. (2017).** Six challenges for neural machine translation. Proceedings of the First Workshop on Neural Machine Translation, Association for Computational Linguistics, Vancouver, pp. 28–39.
15. **Li, Q., Su, H., Niu, C., Wang, D., Li, Z., Feng, S., Zhang, Y. (2019).** Answer-supervised question reformulation for enhancing conversational machine comprehension. Proceedings of the 2nd Workshop on Machine Reading for Question Answering, Association for Computational Linguistics, pp. 38–47.
16. **Li, Y., Su, L., Chen, J., Yuan, L. (2017).** Semi-supervised learning for question classification in CQA. Natural Computing: An International Journal, Vol. 16, No. 4, pp. 567–577.
17. **Liu, J., Yang, Y., Lv, S., Wang, J., Chen, H. (2019).** Attention-based BiGRU-CNN for chinese question classification. Journal of Ambient Intelligence and Humanized Computing.
18. **Mohasseb, A., Bader-El-Den, M., Cocea, M. (2018).** Question categorization and classification using grammar based approach. Information Processing & Management, Vol. 54, No. 6, pp. 1228–1243.
19. **Mohd, M., Hashmy, R. (2018).** Question classification using a knowledge-based semantic kernel. **Pant, M., Ray, K., Sharma, T. K., Rawat, S., Bandyopadhyay, A.**, editors, Soft Computing: Theories and Applications, Springer Singapore, Singapore, pp. 599–606.
20. **Padró, L., Stanilovsky, E. (2012).** FreeLing 3.0: Towards wider multilinguality. Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12), European Language Resources Association (ELRA), pp. 2473–2479.
21. **Ramprasath, M., Hariharan, S. (2012).** Improving qa performance through semantic reformulation. 2012 Nirma University International Conference on Engineering (NUiCONE), pp. 1–4.
22. **Reinsel, D., Gantz, J., Rydning, J. (2018).** The Digitalization of the Word from Edge to Core. `https://www.seagate.com/files/www-content/our-story/trends/files/idc-seagate-dataage-whitepaper.pdf`.
23. **Sewak, M., Karim, M. R., Pujari, P. (2018).** Practical Convolutional Neural Networks: Implement Advanced Deep Learning Models Using Python. Packt Publishing.
24. **Sidorov, G. (2013).** Contrucción no-lineal de n-gramas en la lingüística computacional. SMIA.
25. **Sidorov, G. (2013).** Non-continuous syntactic n-grams. Polibits, Vol. 48, No. 1, pp. 67–75.
26. **Sidorov, G. (2019).** Syntactic n-grams in computational lingusitics. Springer.
27. **Tayyar Madabushi, H., Lee, M. (2016).** High accuracy rule-based question classification using question syntax and semantics. Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, The COLING 2016 Organizing Committee, Osaka, Japan, pp. 1220–1230.
28. **Tayyar Madabushi, H., Lee, M., Barnden, J. (2018).** Integrating question classification and deep learning for improved answer selection. Proceedings of the 27th International Conference on Computational Linguistics, Association for Computational Linguistics, Santa Fe, New Mexico, USA, pp. 3283–3294.
29. **Umamehaswari, M., Ramprasath, M., Hariharan, S. (2012).** Improved question answering system by semantic refomulation. Fourth International Conference on Advanced Computing (ICoAC), pp. 1–4.

30. **Walton, D. (1999).** The fallacy of many questions: On the notions of complexity, loadedness and unfair entrapment in interrogative theory. Argumentation, Vol. 13, pp. 379–383.
31. **Wasim, M., Mahmood, W., Asim, M. N., Khan, M. U. (2019).** Multi-label question classification for factoid and list type questions in biomedical question answering. IEEE Access, Vol. 7, pp. 3882–3896.
32. **Xu, D., Jansen, P., Martin, J., Xie, Z., Yadav, V., Tayyar Madabushi, H., Tafjord, O., Clark, P. (2020).** Multi-class hierarchical question classification for multiple choice science exams. Proceedings of the 12th Language Resources and Evaluation Conference, European Language Resources Association, Marseille, France, pp. 5370–5382.
33. **Xue, X., Tao, Y., Jiang, D., Li, H. (2012).** Automatically mining question reformulation patterns from search log data. Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), Association for Computational Linguistics, pp. 187–192.

Article received on 06/05/2020; accepted on 12/09/2020.
Corresponding author is Noé Alejandro Castro Sánchez.