Journal Homepage: -www.journalijar.com

INTERNATIONAL JOURNAL OF ADVANCED RESEARCH (IJAR)

Article DOI:10.21474/IJAR01/15343
DOI URL: http://dx.doi.org/10.21474/IJAR01/15343

RESEARCH ARTICLE

HYBRID MODEL FOR THE CLASSIFICATION OF QUESTIONS EXPRESSED IN NATURAL LANGUAGE

Sangare Seydou[1,2], Konan Marcellin Brou[1,2], Kouame Appoh[1,2] and Kouadio Prosper Kimou[1,2]

1. Ecole Doctorale Polytechnique de l'INP-HB Yamoussoukro, Côte d'Ivoire.
2. Laboratoire de Recherche en Informatique et Telecommunication (LARIT).

Manuscript Info

Manuscript History
Received: 05 July 2022
Final Accepted: 09 August 2022
Published: September 2022

Key words:-
Machine Learning, Natural Language, Questions Classification, Question-answering system, SPARQL.

Abstract

Question-answering systems rely on an unstructured text corpora or a knowledge base to answer user questions. Most of these systems store knowledge in multiple repositories including RDF. To access this type of repository, SPARQL is the most convenient formal language. It is a complex language, it is therefore necessary to transform the questions expressed in natural language by users into a SPARQL query. As this language is complex, several approaches have been proposed to transform the questions expressed in natural language by users into a SPARQL query.However, the identification of the question type is a serious problem. Questions classification plays a potential role at this level. Machine learning algorithms including neural networks are used for this classification. With the increase in the volume of data, neural networks better perform than those obtained by machine learning algorithms, in general. That is, neural networks, machine learning algorithms also remain good classifiers. For more efficiency, a combination of convolutional neural network with these algorithms has been suggested in this paper. The BICNN-SVM combination has obtained good score not only with small dataset with a precision of 96.60% but also with a large dataset with 94.05%.

Introduction:-

A question-answering system is a system that gives a precise answer to a question asked by a user in natural language. This kind of system can relies on an unstructured text corpora or on a knowledge base to answer the users' questions [1]. An ontological knowledge base encompasses two components: the Tbox and Abox[2]. The Tbox stores the lexicon or ontology of the domain, meta-information that describes the information stored in the Abox. The Abox or fact base stores knowledges of the domain in an RDF repository [3]. To reach this type of repository, SPARQL is the most suitable formal language. However, this complex language can only be decoded by experts. This urges us to transform natural language question into a SPARQL query[4]. This task is performed in several steps: question analysis, expression mapping, disambiguation, and construction of the corresponding SPARQL query. Concerning the analysis of the question, most of the systems[5],[6], [7]use the parsers for the identification of the dependencies between words in order to characterize the question and the classified.These linguistic tools often make mistakes that are disseminated through the conversion process[7],[8]. In[8]the researchers have proposed a set of syntactic heuristics to do the conversion, however its analysis is very slow, it is even impossible to identify the questions starting with when. There are low precision and recall for the questions based on theaggregation.This

CorrespondingAuthor:- Sangare Seydou
Address:- Ecole Doctorale Polytechnique de l'INP-HB Yamoussoukro Côte d'Ivoire.

is why for[9]the major challenges at the level of the analysis of the question are: the identification of the type of thequestion, multilingualism and identification of the operators of aggregation, comparison and negation.In addition, the questions are identified in categoriesaccording to the taxonomy of[10]which gives no information about the type of SPARQL query corresponding to the natural language of the user.

Contribution

In the framework of this study, animproved version of LC_QUAD2.0 has been proposed where I classify natural language questions based on the triplet patterns of SPARQL queries. Then for the identification of the type of question, a hybrid model of question classification has been proposed: BICNN-SVM.Support Vector Machine (SVM)is a powerful classifier.However, it needs the lexical, syntactic and semantic characteristics of each question to be trained. Extracting these features is time-consuming and requires expert knowledge. The extraction of features by the convolutional neural network CNN which allows to automatically extract the features of each question and transmit them to the SVM classifier has been proposed. To take into account the context and position of each word, every question will be traversed by CNN filters in two directions: from the beginning to the end and from the end to the beginning. This is why, it is known as BICNN.

This paper is structured as follows: section II is related literature review on the questions classification. In section III, irregularities on the LC_QUAD2.0 dataset. In section IV, a class proposal for LC_QUAD2.0 description of the proposed model.In section V, evaluation of the model.

Related Works:-

According to[11]there are four main approaches to question classification which are: (1) rule-based approaches, (2) machine learning approaches, (3) deep learning approaches, (4) hybrid approaches.

Rule-Based Approaches[12], [13], [14], [15]rely on large-scale predefined grammatical, syntactic and semantic rules to determine the type of question. These rules are done manually. This process requires a lot of time and expertise. For [16] these approaches can hardly be extended to the other domains. They are less performant than approaches that are based on machine learning.

Machine learning is based on several methods, among which: Support Vector Machine (SVM), Naïve Bayes, Decision Tree, K Nearest Neighbor, Random Forest. [17] Extracts features such as question focus, question length, named entities, lexical block. It then uses them for question classification with maximum entropy. [18]Uses a function called kernel tree to extract the syntactic features of the questions and SVM to do question classification.Without using syntactic and semantic features [19]proposes a tree core with SVM to identify the questions. It has achieved an accuracy of 87.4%. [20]analyzes the impact of instance selection on k-nearest neighbor (KNN) based text classification. [21]uses four classifiers that are: Naïve Bayes, J48, BFTree and OneR for sentiment analysis optimization. In terms of well-ranked instances, OneR is the top performer. According to[16] the traditional machine learning methods provide good results. However, they do not perform well with large datasets or with semantically complex content. These methods use Term Frequency-Inverse Document Frequency (TF-IDF) for the representation and vectorization of the texts' characteristics.Whereas this type of feature representation does not grasp the meaning of words or identify relationships between them[22]. In fact, word embedding is an alternative that quantifies the semantics between words. This type of feature representation is generally used in approaches based on deep learning which have better performances than the traditional methods. [22]uses Long Short Term Memory (LSTM) and Bidirectional LSTM (BILSTM) for the classification of Arabic questions in the field of health.

[23]was the first to use the convolutional neural network for natural language processing.[24]proposes a convolutional network with dynamic k-max-pooling for question and sentiment classification. Dynamic k-max-pooling generates a feature map of each sentence. It captures short and long relationships between words. The parameter k of the pooling is chosen according to the size of the sentence. In order to improve the performance of convolutional neural networks for sentence classification [25]adds machine learning strategies such as normalization.For the prediction of small classes to be more accurate[26]proposes a model with an extended structure of CNN

Analysis ofthe LC-QuAD2.0 DataSet:-

[1] and [27]classify questions according to the different kind of requests for the LC-QUAD2.0 dataset. However, [27]stresses irregularities in the data set. Accordingly, an overview of the three fields which are: subgraph,

template_id, template has been done to get table1. It has been found that each of the subgraphs center and right-subgraph contain two different templates, rank and boolean-one-hop-right-subgraph containing three different template. Statement_property and boolean_double_one_hop_right_subgraph also contain four different templates. Confusion in the template_id field is noticeable. For example in the rank, two different templates have identical template_id rank_2. The same problem is observable at the level of boolean-one-hop-right-subgraph and boolean-double-one-hop-right-subgraph. In the dataset there are 1556 questions that have neither template nor template_id. Therefore, they are not classified in any subgraph. These irregularities can have drawbacks on the performance of the classification.

Proposed Model:-

This section, presenting my approach to classifying natural language questions is structured in four parts that are: (1) The proposed taxonomy; (2) Transformation of questions; (3) The convolutional neural network; (4) Support Vector Machine (SVM).

$\mathcal{D}$, a dataset comprising several questions:

$\mathcal{D} = \{(q_i, y_j)\}$where$i \in \{1, 2, \dots, n\}$ et $j \in \{1, 2, \dots, m\}$

with: q_i the ith natural language question and y_j the j^{th}class that corresponds to the type of SPARQL query pattern triplet used to extract q_i 's answer from the knowledge base.

For example given a question q_i : Who is the film editor and director of Reservoir Dogs? Classification will allow to identify corresponding y_j : (ent-pred-obj1. ent-pred-obj2)

Proposed Taxonomie

The overview of the dataset has allowed to see that it contains 25 templates. Each template corresponds to a set of questions. Each template can be assigned a unique identifier to as presented in table2. The questions have be classified according to the 25 templates.

Features Extraction

Transforming of questions

The question classification approach is based on Convolutional Neuronal Network (CNN). It cannot process texts directly. The questions are therefore transformed into digital form.This task is carried out by the word embedding which makes it possible to transform each word of a dictionary into a vector of numbers.There are several types of word embedding. But Glove is used throught this research because it is simple and easy to implement.

A question q_i is considered as a sequence ofl words:

$$q_i = [m_1 m_2 m_3 \dots m_l] \quad (1)$$

Each word $m_i$is transformed using the word embedding into a vector$v_i \in \mathbb{R}^k$of dimensionk. These vectors are concatenated to give a matrix representation$MP_i$of the i^{th} question.

$$MP_i = v_1 \oplus v_2 \oplus v_3 \oplus \dots \oplus v_l \quad (2)$$

WhereMP_i is the matrix of dimension $\times k$, which represents the question, $\oplus$ is the concatenation operator.

Convolution Layer

The convolution layer receives as input the matrixMP_i. The objective at this layer is to identify a set of characteristics, proper to each type of question. With convolutional neural networks, the characteristics of each type of question are not predefined, but learned automatically.Convolution operation consisting in sliding a filter $F_i \in \mathbb{R}^{t \times k}$to a window of t word vectors of MP_i is carried out. During the convolution operation, by varying the size t, different filters can be used to cover several word vectors of the matrix MP_i. In this work the filters are slid in two directions of the matrixMP_ias proposed by [28]. From the start to the end and from the end to the start.Each convolution help obtain a new characteristic$\overrightarrow{c_i}$and $\overleftarrow{c_i}$of the question by:

$$\overrightarrow{c_i} = f\left(\overrightarrow{F_i}. MP_i + b_i\right) \quad (3)$$

$$\overleftarrow{c_i} = f\left(\overleftarrow{F_i}. MP_i + b_i\right) \quad (4)$$

Where$\overrightarrow{F_i}$is the i^{th} filter that slides along the matrix$MP_i$from start to end.

$\overleftarrow{F_i}$is the i^{th} filter that slides from the end of the matrix to the beginning, $b_i$is the bias andfis the nonlinear activation function.

Each filter provides a characteristic card:

$$\overrightarrow{c_i} = [\overrightarrow{c_1}, \overrightarrow{c_2}, \dots, \overrightarrow{c_{1-n+1}}] \quad (5)$$

$$\overleftarrow{c_i} = [\overleftarrow{c_1}, \quad \overleftarrow{c_2}, \dots, \quad \overleftarrow{c_{1-n+1}}] \quad (6)$$

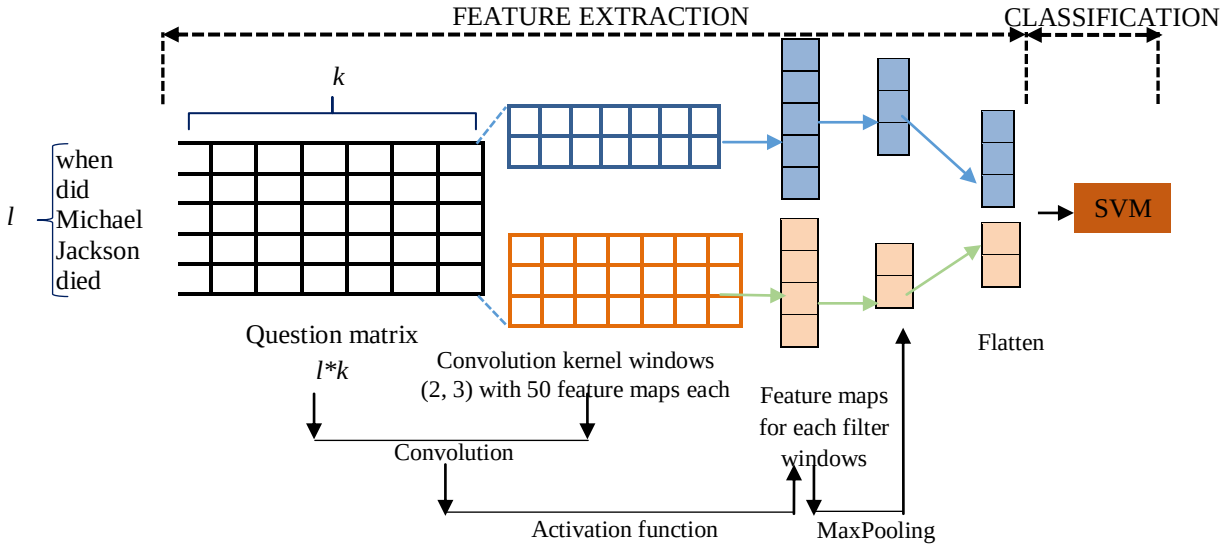

Fig.1:- architecture of the proposed model.

Pooling Layer

At this layer, the Maxpooling operation is applied to each $\vec{c}$ and $\overleftarrow{c}$to choose the maximum value.The goal is to find out the most important features generated by a filter.

$$\widehat{\vec{c}_i} = \max\{\vec{c}_i\} \quad (7)$$

$$\widehat{\overleftarrow{c}_i} = \max\{\overleftarrow{c}_i\} \quad (8)$$

$$\hat{\vec{c}} = [\widehat{\vec{c}_1}, \quad \widehat{\vec{c}_2}, \dots, \widehat{\vec{c}_r}] \quad (9)$$

$$\hat{\overleftarrow{c}} = [\widehat{\overleftarrow{c}_1}, \quad \widehat{\overleftarrow{c}_2}, \dots, \widehat{\overleftarrow{c}_r}] \quad (10)$$

Before flatten a concatenation of $\hat{\vec{c}} \oplus \hat{\overleftarrow{c}}$is done.

$$x_i = \hat{\vec{c}} \oplus \hat{\overleftarrow{c}} = [\widehat{\vec{c}_1}, \quad \widehat{\vec{c}_2}, \dots, \widehat{\vec{c}_r}, \widehat{\overleftarrow{c}_1}, \quad \widehat{\overleftarrow{c}_2}, \dots, \widehat{\overleftarrow{c}_r}] \quad (11)$$

Where$x_i$represents the characteristics of thei^{th}question which are sent to the SVM classifier.

Classifier Support Vector Machine (SVM) Multi-Class

SVM is a maximum margin classifier developed by [29].The main idea is to search for an optimal hyperplane separating the data into two sets.It is a binary classifier.There are several methods to adapt it to multi-class cases: one against one, one against all, Oriented Acyclic Graph and the descending hierarchical method [30]. In our work, we use the «one against all» method. Because according to [31] and [32], it is robust, simple to implement and able to manage large-scale datasets.This method consists in finding a hyperplane$p_k(w_k, b_k)$for each class $y_k$in order to separate it from all the others [32].The class $y_k$is considered positive and the others negative.Which requires k binary SVM for a problem of k classes.[33]presents the optimization problem as follows:

Where w weight vector, b the bias, ϕ the transformation function in a higher dimension, C makes it possible to control the trade-off that may exist between maximizing the margin and minimizing the classification error and ξ deviation variable.

The resolution of (12) allows k hyperplanes which represent the decision functions:

$$\begin{cases} (w^1)^T \phi(x_j) + b^1 \\ \vdots \\ (w^k)^T \phi(x_j) + b^k \end{cases} \quad (13)$$

The class of element x is the largest value of the decision function:

$$class(x) = arg\,max_{i=1,\dots,k}\left((w^i)^T \phi\left(x_j\right) + b^i\right) \quad (14)$$

Table 1:- Overview of LC-QUAD2.0

N°	TEMPLATE	TEMPLATE_ID	SUBGRAPH	NUMBER OF QUESTIONS
1	E REF ?F	1.1	center	3304
2	?D RDE E	1.2	center	740
3	(E pred ?Obj) prop value	statement_property_1	statement_property	2969
4	(E pred F) prop ?value	statement_property_2	statement_property	2943
5	Count Obj (ent-pred-obj)	Count_1	statement_property	656
6	Count ent (ent-pred-obj)	Count_2	statement_property	768
7	E REF ?F . ?F RFG G	1	right-subgraph	2505
8	E REF xF .xF RFG ?G	2	right-subgraph	1923
9	C RCD xD .xD RDE ?E	5	left-subgraph	1791
10	<?S P O ; ?S InstanceOf Type>	2	simple question left	2042
11	<S P ?O ; ?O instanceOf Type>	1	simple question right	1872
12	ASK ?sbj ?pred ?obj filter ?obj = num	3	booleanwithfilter	1672
13	[]			1556
14	<?S P O ; ?S instanceOf Type ; starts with character >	2	string matching simple contains word	1307
15	<?S P O ; ?S instanceOf Type ; contains word >	3	string matching type + relation contains word	1307
16	select where (ent-pred-obj1 . ent-pred-obj2)	1	two intentions right subgraph	740
17	?Eis_a Type. ?Epred Obj. ?E-secondClause value. MAX (value)	Rank_2	Rank	377
18	?Eis_a Type. ?Epred Obj. ?E-secondClause value. MIN (value)	Rank_2	Rank	377
19	?Eis_a Type, ?E predObj value. MAX/MIN (value)	Rank_1	Rank	377
20	Ask (ent-pred-obj)	1	booleanone_hop right subgraph	318
21	Ask (ent-pred-obj`)	1	booleanone_hop right subgraph	173
22	Ask (ent`-pred-obj)	1	booleanone_hop right subgraph	9
23	Ask (ent-pred-obj1 . ent-pred-obj2)	2	boolean double one_hop right subgraph	212
24	Ask (ent-pred-obj1` . ent-pred-obj2)	2	boolean double one_hop right subgraph	147
25	Ask (ent-pred-obj1 . ent-pred-obj2`)	2	boolean double one_hop right subgraph	125
26	Ask (ent`-pred-obj1 . ent`-pred-obj2)	2	boolean double one_hop right subgraph	16
	TOTAL			30226

Table 2:- Class Proposed.

N°	TEMPLATE_ID	ID	TEMPLATE	NUMBER OF QUESTIONS
1	Center_11	CT1	E REF ?F	3304
2	Center_12	CT2	?D RDE E	740
3	simple question left_2	SQL	<?S P O ; ?S InstanceOf Type>	2042
4	simple question right_1	SQR	<S P ?O ; ?O instanceOf Type>	1872
5	right-subgraph_1	RS1	E REF ?F . ?F RFG G	2505
6	right-subgraph_2	RS2	E REF xF .xF RFG ?G	1923
7	left-subgraph_5	LSB	C RCD xD .xD RDE ?E	1791
8	statement_property_1	SP1	(E pred ?Obj) prop value	2969
9	statement_property_2	SP2	(E pred F) prop ?value	2943
10	statement_property_Count_1	CU1	Count Obj (ent-pred-obj)	656
11	statement_property_Count_2	CU2	Count ent (ent-pred-obj)	768
12	Rank_1	RK1	?Eis_a Type, ?E predObj value. MAX/MIN (value)	377
13	Rank_2_MIN	RK2	?Eis_a Type. ?Epred Obj. ?E-secondClause value. MIN (value)	377
14	Rank_2_MAX	RK3	?Eis_a Type. ?Epred Obj. ?E-secondClause value. MAX (value)	377
15	Two_intentions_right_subgraph_1	TIR	select where (ent-pred-obj1 . ent-pred-obj2)	740
16	String_matching_simple_contains_word_2	ST1	<?S P O ; ?S instanceOf Type ; starts with character >	1307

17	String_matching_type_relation_contains_word_3	ST2	<?S P O ; ?S instanceOf Type ; contains word >	1307
18	Boolean_one_hop_right_subgraph_1	BO1	Ask (ent-pred-obj)	318
19	Boolean_one_hop_right_subgraph_2	BO2	Ask (ent-pred-obj`)	173
20	Boolean_one_hop_right_subgraph_3	BO3	Ask (ent`-pred-obj)	9
21	Boolean_double_one_hop_right_subgraph_1	BD1	Ask (ent-pred-obj1 . ent-pred-obj2)	212
22	Boolean_double_one_hop_right_subgraph_2	BD2	Ask (ent-pred-obj1` . ent-pred-obj2)	147
23	Boolean_double_one_hop_right_subgraph_3	BD3	Ask (ent-pred-obj1 . ent-pred-obj2`)	125
24	Boolean_double_one_hop_right_subgraph_4	BD4	Ask (ent`-pred-obj1 . ent`-pred-obj2)	16
25	booleanwithfilter	BFL	ASK ?sbj ?pred ?obj filter ?obj = num	1672
	TOTAL			28670

Experimental Study:-
In this section, the performance of the proposed model is evaluated on two datasets. It is also compared with those of other question classification models.

Experimental Dataset
To evaluate the model, two datasets has been selected because they are frequently used in questions classification.
- Text REtrieval Conference (TREC): allows to classify a question into 6 types of question. This enables to know if the question is related a person, a place, digital information [10].
- Large-scale Complex Question Answering Dataset (LC-QUAD2.0): is a large data set containing complex questions.

These data sets that are texts have cleaned because they often contain unnecessary characters and words.

Table 3:- Statistics on the two data sets.

Dataset	Classes	Train set	Test set
TREC	6	5454	500
LC-QUAD2.0	25	22962	5741

Parameters of the Model
For experimenting the system, datasets that contain raw words are used. These words must be converted into vectors in order to extract the characteristics of each question. Several methods as proposed by [34] have been referred to:
1. Bag-of-words (BOW): a question is considered as a bag of words. Syntactic structure and semantic relationships between words are ignored.
2. Term frequency-inverse document frequency (TF-IDF): assigns a higher weight to words with a high or low document frequency term without taking into account the similarity between the words.
3. Word embedding: each word in the dataset is mapped to a d-dimensional vector of real numbers taking into account the syntactic and semantic structure of the question. Severaltypes of word embedding already exist. We use pre-trained 300 dimensional Global Vectors for Word Representation (GloVe).For GloVelevel feature extraction, the convolutional neural network is used with four Conv2D convolution layers. In each of the first two layers there are 50 filters of size 2. Each of the last two contains 50 filters of size 3. MaxPooling2D is applied to each of the layers. The four MaxPooling2D are concatenated and then connected to the flattern layer. The weights obtained at the output of flattern are thus sent to machine learning algorithms such Support Vector Machines (SVM), Naive Bayes (NB) and Random Forest (RF).

Results of the Experimental Study
The machine learning algorithms is presented and analysed for three types of extracted features. It is concerned two datasets of different sizes. The evaluation of the models is based on the following metrics: the precision, the recall and the F1-score. The formulas are as follows:

$$precision = \frac{TP}{TP + FP} \quad (15)$$

$$recall = \frac{TP}{TP + TN} \quad (16)$$

$$F1 - score = \frac{2 \times precision \times recall}{precision + recall} \quad (17)$$

Where:
$TP$is 'True Positive'. It is the number of samples predicted correctely.

*FP*is 'False Positive'. It is the number of samples in which the other classifications are incorrectly predicted as this classification.
*TN*is 'True Negative'. It is the number of samples that are incorrectly predicted as other classifications.

TREC DataSet

Table 4:- Results with the TREC DataSet.

Features	Models	Precision (%)	Recall (%)	F1-score (%)
Bag of word	SVM	84.81	84.20	84.32
	Naive Baye	79.69	78.80	79.06
	Random Forest	75.27	74.00	74.44
	Logistic Regression	84.45	84.20	84.32
TD-IDF	SVM	86.06	85.60	85.55
	Naive Baye	80.14	78.40	79.00
	Random Forest	78.43	76.80	77.60
	Logistic Regression	83.96	83.60	83.55
GloVe	BICNN-SVM	96,60	95,78	96,18
	BICNN-Naive Baye	90.81	89.95	90.37
	BICNN-Random Forest	91,60	91,30	91,44
	BICNN-Logistic Regression	90.92	91.00	90,95

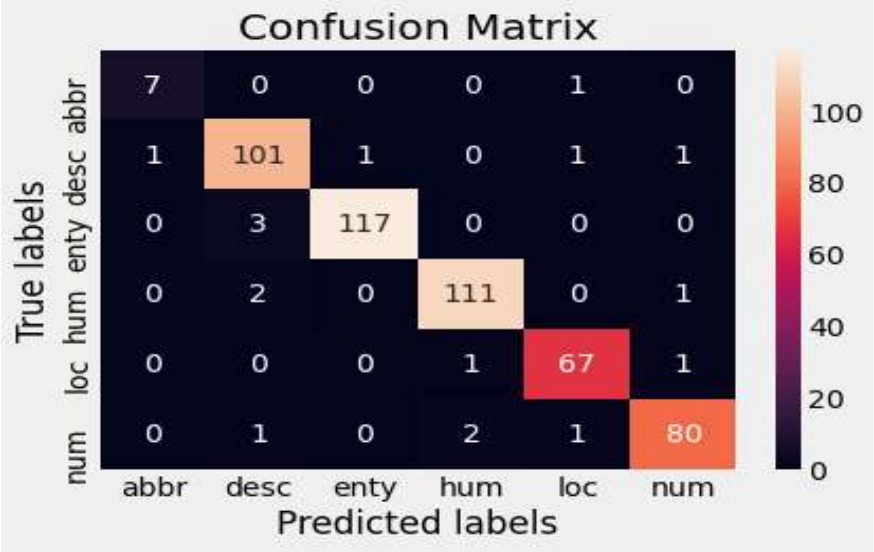

Fig 3:- Confusion Matrix of the BICNN-SVM Model with TREC DataSet.

Table 4 presents the classification performances of the SVM, Naive Baye, Random Forest and Logistic Regression classifiers as a function of the Bag of Word, TD-IDF and Glove features a pre-trained word embedding.With bag of word SVM identifies questions with 84.81% precision while Naive Baye identifies them correctly with 79.69% accuracy, Random Forest 75.27% and Logistic Regression 84.45%. At the level of TD-IDF, SVM makes a classification with a precision of 86.06 while Naive Baye at 80.14%, Random Forest 78.43% and Logistic Regression 83.96%. For the Glove pre-trained word embedding, BICNN-SVM performs well with aprecision of 96.60% while BICNN-Naive Baye has a rate of 90.81%, BICNN-Random Forest 91.60% and BICNN-Logistic Regression 90.92%.

LC_QUAD2.0 DataSet

Table 5:- Results with LC_QUAD2.0 DataSet.

Features	Models	Precision (%)	Recall (%)	F1-score (%)
	SVM	84.18	80.67	81.43
	Naive Baye	81.43	72.91	75.69

Bag of word	Random Forest	80.98	77.45	78.16
	Logistic Regression	83.53	81.83	82.10
TD-IDF	SVM	81.38	79.99	80.32
	Naive Baye	82.40	69.45	73.47
	Random Forest	80.68	78.44	79.54
	Logistic Regression	82.52	79.19	80.16
Pretrained GloVe	BICNN-SVM	94,05	93,65	93,84
	BICNN-Naive Baye	82.71	82.73	82.71
	BICNN-Random Forest	82.97	82.62	82.72
	BICNN-Logistic Regression	83.39	82.66	82.86

Table 6 shows the results of the evaluation of the different models according to the three characteristics Bag of Word, TD-IDF and pre-trained GloVe.

At the level of the Bag of Word SVM obtains a precision of 84.18%, Naive Baye at a precision of 81.43%, Random Forest 80.98% and Logistic Regression 83.53%. With TD-IDF, SVM obtains a precision of 81.38%, Naive Baye at 82.40%, Random Forest 80.68% and Logistic Regression 82.52%. For pre-trained Glove BICNN-SVM obtains a good performance of 94.05% while BICNN-Naive Baye obtains 82.71%, BICNN-Random Forest 82.97% and BICNN-Logistic Regression 83.39%.

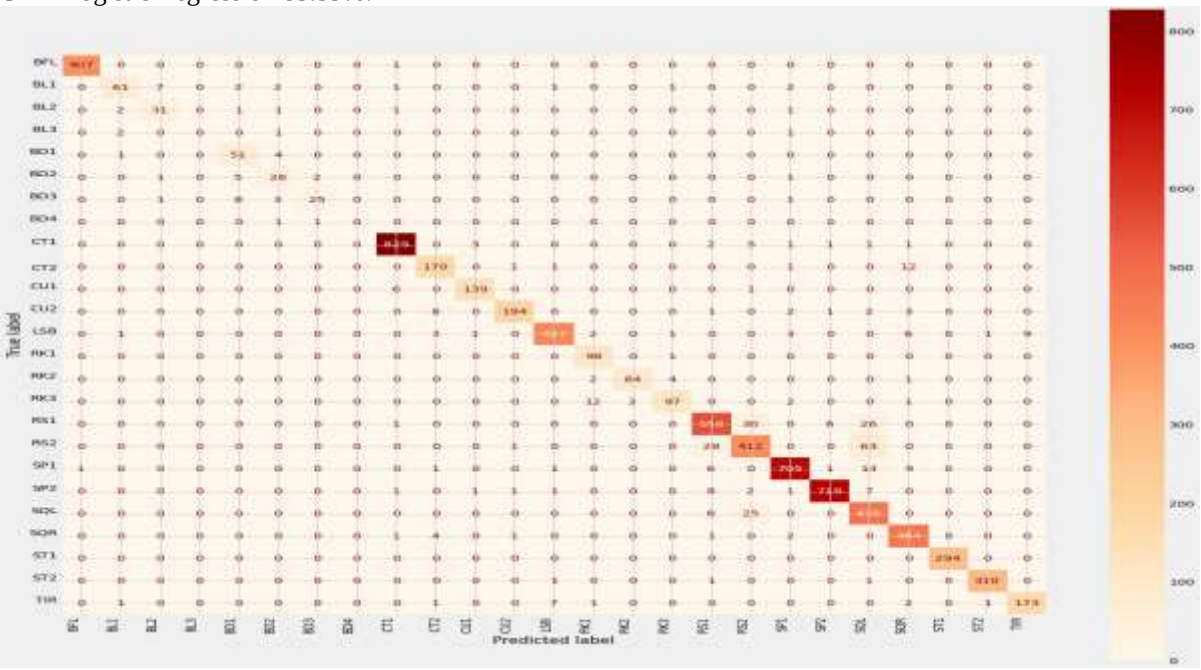

Fig 4:- Confusion Matrix of the BICNN-SVM Model with LC-QUAD2.0 DataSet.

Discussion:-

They obtain better results than the results obtained with the bag of word and TD-IDF characteristics. This good performance is due to the vector representation strength of the pre-trained GloVe word embedding. It also depends on the extraction power of the BICNN network and its ability to keep the link between words and their order in two ways to understand the context. For the vector representation of the words glove takes into account not only the frequency of each word but also the relation between each word and the other components of the question. In addition, it should also be noted that whatever the characteristics, the results of SVM are better than those of the other algorithms. The results obtained with the TREC dataset are also better than those of the LC_QUAD2.0 set, because LC_QUAD2.0 is larger than TREC. These results show that it is possible to improve the performance of the machine learning algorithms by helping them benefit from the vector representation strength of word embeddings and the extraction power of neural networks.

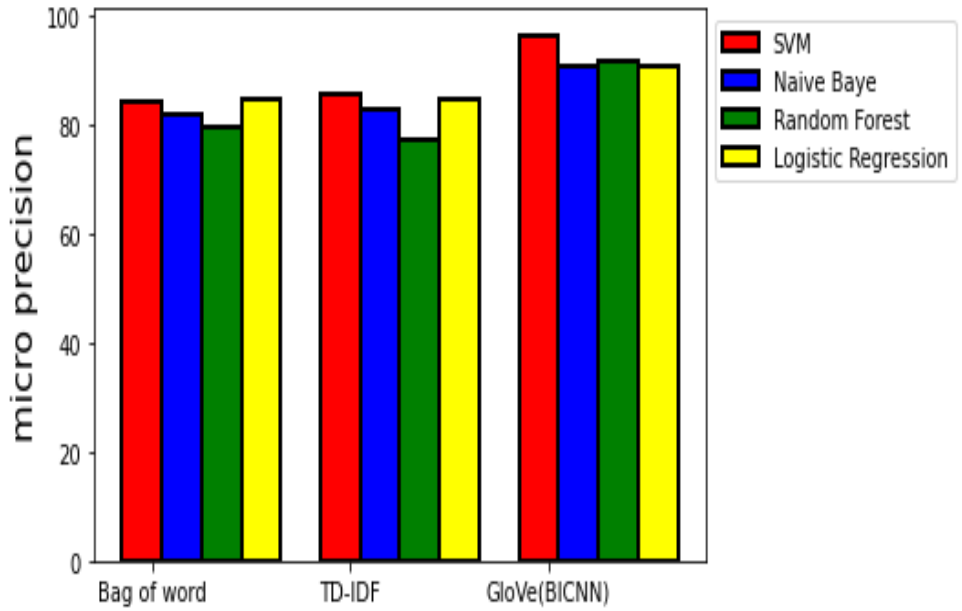

Fig 5:- Accuracy of Classifiers with TREC.

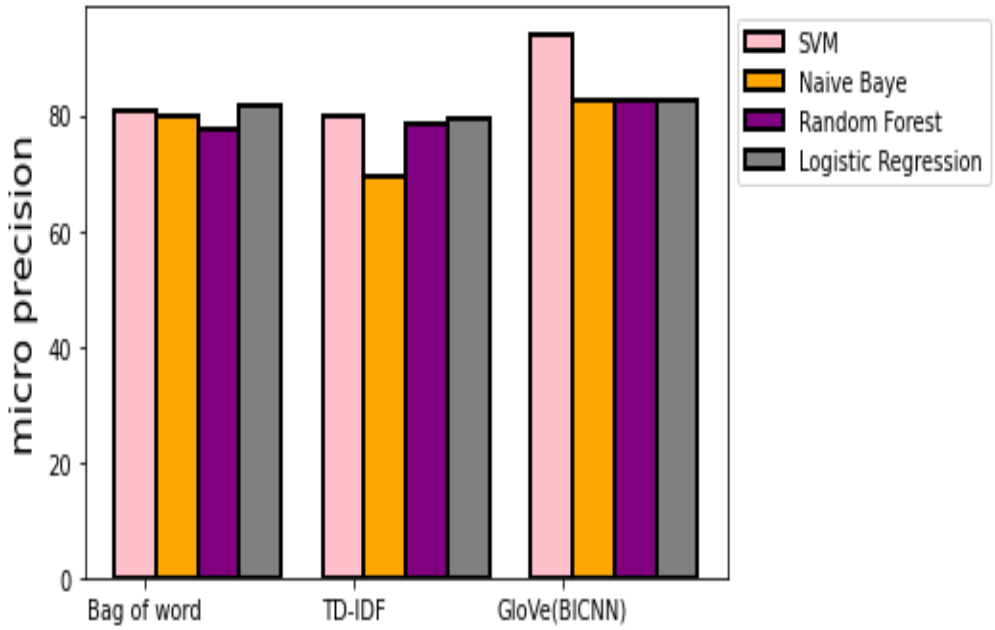

Fig 6:- Accuracy of Classifiers with LC_QUAD2.0.

Conclusion:-

In this work, a version of LC_QUAD2.0 has been proposed. The natural language questions based on the triplet patterns of SPARQL queries have been classified. A combination of neural network with machine learning algorithms for question classification have also been referred to. The Combined models better perform with both datasets. The BICNN-SVM model achieves an accuracy of 96.60% with the TREC dataset and 94.05% with the large LC_QUAD2.0 dataset.

In the future, a study on the effect of other word embeddings such as FastText, ELMO, BERT on the performance of the hybrid model will be carried out.

References:-

[1] D. A. Evseev et M. Yu. Arkhipov, « SPARQL query generation for complex question answering with BERT and BiLSTM-based model », p. 13, juin 2020.

[2] F. Baader, D. Calvanese, D. McGuinness, P. Patel-Schneider, et D. Nardi, The Description Logic Handbook: Theory, Implementation and Applications. Cambridge University Press, 2003.

[3] M. A. Paredes-Valverde, M. Á. Rodríguez-García, A. Ruiz-Martínez, R. Valencia-García, et G. Alor-Hernández, « ONLI: An ontology-based system for querying DBpedia using natural language paradigm », Expert Systems with Applications, vol. 42, no 12, p. 5163‑5176, juill. 2015, doi: 10.1016/j.eswa.2015.02.034.

[4] D. Diefenbach, « Question answering over Knowledge Bases », p. 215, 2018.

[5] K. Xu, S. Zhang, Y. Feng, et D. Zhao, « Answering Natural Language Questions via Phrasal Semantic Parsing », in Natural Language Processing and Chinese Computing, vol. 496, C. Zong, J.-Y. Nie, D. Zhao, et Y. Feng, Éd. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014, p. 333‑344. doi: 10.1007/978-3-662-45924-9_30.

[6] S. He, Y. Zhang, K. Liu, et J. Zhao, « CASIA@V2: A MLN-based Question Answering System over Linked Data », p. 11, 2014.

[7] A. Abujabal, M. Yahya, M. Riedewald, et G. Weikum, « Automated Template Generation for Question Answering over Knowledge Graphs », in Proceedings of the 26th International Conference on World Wide Web, Perth Australia, avr. 2017, p. 1191‑1200. doi: 10.1145/3038912.3052583.

[8] P. Ochieng, « PAROT: Translating natural language to SPARQL », Expert Systems with Applications: X, vol. 5, p. 100024, avr. 2020, doi: 10.1016/j.eswax.2020.100024.

[9] D. Diefenbach, V. Lopez, K. Singh, et P. Maret, « Core techniques of question answering systems over knowledge bases: a survey », KnowlInfSyst, vol. 55, no 3, p. 529‑569, juin 2018, doi: 10.1007/s10115-017-1100-y.

[10] X. Li et D. Roth, « Learning question classifiers », in Proceedings of the 19th international conference on Computational linguistics -, Taipei, Taiwan, 2002, vol. 1, p. 1‑7. doi: 10.3115/1072228.1072378.

[11] M. Zulqarnain, A. KhalafZagerAlsaedi, R. Ghazali, M. G. Ghouse, W. Sharif, et N. Aida Husaini, « A comparative analysis on question classification task based on deep learning approaches », PeerJ Computer Science, vol. 7, p. e570, août 2021, doi: 10.7717/peerj-cs.570.

[12] E. Hovy, L. Gerber, U. Hermjakob, C.-Y. Lin, et D. Ravichandran, « Toward semantics-based answer pinpointing », in Proceedings of the first international conference on Human language technology research - HLT '01, San Diego, 2001, p. 1‑7. doi: 10.3115/1072133.1072221.

[13] B. Magnini, M. Negri, R. Prevete, et H. Tanev, « Mining Knowledge from Repeated Co-occurrences: DIOGENE at TREC-2002 », p. 9, 2002.

[14] G. Nolano, M. F. Elahi, M. P. di Buono, B. Ell, et P. Cimiano, « An Italian Question Answering System Based on Grammars Automatically Generated from Ontology Lexica », p. 6.

[15] B. Galitsky, « Matching parse thickets for open domain question answering », Data & Knowledge Engineering, vol. 107, p. 24‑50, janv. 2017, doi: 10.1016/j.datak.2016.11.002.

[16] J. Liu, Y. Yang, S. Lv, J. Wang, et H. Chen, « Attention-based BiGRU-CNN for Chinese question classification », J Ambient Intell Human Comput, juin 2019, doi: 10.1007/s12652-019-01344-9.

[17] K. Kocik, « Question Classification using Maximum Entropy Models », p. 99.

[18] D. Zhang et W. S. Lee, « Question Classification using Support Vector Machines », p. 7.

[19] X. Huang, A. Maier, J. Hornegger, et J. A. K. Suykens, « Indefinite kernels in least squares support vector machines and principal component analysis », Applied and Computational Harmonic Analysis, vol. 43, no 1, p. 162‑172, juill. 2017, doi: 10.1016/j.acha.2016.09.001.

[20] F. Barigou, « Impact of Instance Selection on kNN-Based Text Categorization », Journal of Information Processing Systems, vol. 14, no 2, p. 418‑434, avr. 2018, doi: 10.3745/JIPS.02.0080.

[21] J. Singh, S. Gurvinder, et S. Rajinder, « Optimization of sentiment analysis using machine learning classifiers », Hum. Cent. Comput. Inf. Sci., vol. 7, no 1, p. 32, déc. 2017, doi: 10.1186/s13673-017-0116-3.

[22] H. Faris, M. Habib, M. Faris, A. Alomari, P. A. Castillo, et M. Alomari, « Classification of Arabic healthcare questions based on word embeddings learned from massive consultations: a deep learning approach », J Ambient Intell Human Comput, vol. 13, no 4, p. 1811‑1827, avr. 2022, doi: 10.1007/s12652-021-02948-w.

[23] R. Collobertet J. Weston, « A Unified Architecture for Natural Language Processing: Deep Neural Networks with Multitask Learning », p. 8, 2011.

[24] N. Kalchbrenner, E. Grefenstette, et P. Blunsom, « A Convolutional Neural Network for Modelling Sentences », arXiv:1404.2188 [cs], avr. 2014, Consulté le: 1 septembre 2020. [En ligne]. Disponible sur: http://arxiv.org/abs/1404.2188

[25] Y. Kim, « Convolutional Neural Networks for Sentence Classification », arXiv:1408.5882 [cs], sept. 2014, Consulté le: 19 septembre 2020. [En ligne]. Disponible sur: http://arxiv.org/abs/1408.5882

[26] P. R. Dachapally et S. Ramanam, « In-depth Question classification usingConvolutional Neural Networks », p. 4, 2017.

[27] A. K. Dileep, A. Mishra, R. Mehta, S. Uppal, J. Chakraborty, et S. K. Bansal, « Template-based Question Answering analysis on the LC-QuAD2.0 Dataset », in 2021 IEEE 15th International Conference on Semantic Computing (ICSC), Laguna Hills, CA, USA, janv. 2021, p. 443‑448. doi: 10.1109/ICSC50631.2021.00079.

[28] T. Lei, Z. Shi, D. Liu, L. Yang, et F. Zhu, « A novel CNN-based method for Question Classification in Intelligent Question Answering », in Proceedings of the 2018 International Conference on Algorithms, Computing and Artificial Intelligence - ACAI 2018, Sanya, China, 2018, p. 1‑6. doi: 10.1145/3302425.3302483.

[29] C. Cortes et V. Vapnik, « Support-vector networks », Mach Learn, vol. 20, no 3, p. 273‑297, sept. 1995, doi: 10.1007/BF00994018.

[30] T. Guernine, Classification hiérarchique floue basée sur le SVM et son application pour la catégorisation des documents. Ottawa: Library and Archives Canada = Bibliothèqueet Archives Canada, 2011.

[31] R. Rifkin et A. Klautau, « In Defense of One-Vs-All Classification », p. 41.

[32] F. Perronnin, Z. Akata, Z. Harchaoui, et C. Schmid, « Towards good practice in large-scale learning for image classification », in 2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, juin 2012, p. 3482‑3489. doi: 10.1109/CVPR.2012.6248090.

[33] V. N. Vapnik, « An overview of statistical learning theory », IEEE Trans. Neural Netw., vol. 10, no 5, p. 988‑999, sept. 1999, doi: 10.1109/72.788640.

[34] K. Kowsari, « kk7nc/Text_Classification ». 5 novembre 2021. Consulté le: 5 novembre 2021. [En ligne]. Disponible sur: https://github.com/kk7nc/Text_Classification.