iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 11 **Issue:** XII **Month of publication:** December 2023

DOI: https://doi.org/10.22214/ijraset.2023.57252

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 11 Issue XII Dec 2023- Available at www.ijraset.com

A Fuzzification-Based Processing of Natural Language Query Using Database

Dr.Madhav M. Bokare[1], Mr.Amol V. Suryawanshi[2]
Associate Professor Department of Computer Science,SSBES ITM College,Nanded,

Abstract: ***The requirement for natural language interfaces to database has turn out to be more and more acute as increasingly more people get entry to data from internet browsers. But NLI (Natural Language Interface) is only usable if they map natural language inquiries to sql queries effectively Natural Language processing is turning into one of the maximum energetic areas in Human-pc interaction. It is a department of AI (artificial Intelligence) which incorporates in records retrieval, gadget translation and Language evaluation. The intention of NLP (Natural Language Processing) is to enable communique between human beings and computers without requiring to memorization of complicated commands and techniques. In other words NLP (natural Language Processing) is strategies that may make the pc apprehend the languages evidently utilized by humans. The main motive of Natural Language query Processing is for an English sentence to be interpreted by way of the pc and suitable action taken. Asking questions to databases in Natural language is a totally handy and easy approach of statistics access, in particular for casual customers who do not recognize complicated database question languages which includes square.***

Keywords: NLI(Natural Language Interface), AI(Artificial Intelligence), NLP(Natural Language Processing), NLIDB(Natural Language Interface to Database)

I. INTRODUCTION

Natural language processing (NLP) is the ability of a computer program to recognize human language as it's far spoken and written known as Natural language. It is a component of synthetic intelligence (AI). NLP permits computers to apprehend Natural language as people do. whether or not the language is spoken or written, Natural language processing makes use of artificial intelligence to take actual-world enter, process it, and make sense of it in a way a pc can apprehend. Simply as people have exceptional sensors along with ears to pay attention and eyes to peer computer systems have packages to read and microphones to collect audio. And just as human beings have a brain to manner that input, computer systems have a program to process their respective inputs. Sooner or later in processing, the input is converted to code that the pc can recognize.

A database management device (DBMS) is middleware that allows programmers, database directors (DBAs), software packages and cease users to save, arrange, get right of entry to, query and manage facts in a database. DBMSs are crucial because they provide green and reliable mechanisms for organizing, dealing with and using enormous amounts of information whilst additionally ensuring facts integrity and offering different facts control benefits. Within the agency, database control structures offer database administrators (DBAs) with an established framework that enables facts sharing among specific departments, groups and programs. The DBMS affords employees with managed and organized get admission to information that they can use to drive innovation and help their company hold an aggressive side.

NLIDB (Natural Language Interface to Database) is a way that could make the laptop understand the languages obviously used by people, but now not with the aid of artificial or man-made language including a programming language.

The main purpose is asking inquiries to databases in natural language which is a very convenient and clean approach of records get entry to, especially for casual customers who do no longer recognize complex database question languages consisting of square.

To paintings with any RDBMS one need to recognize the syntax of the instructions of that precise database software program (Microsoft square. Oracle, and many others.),

1) Here the natural language processing is accomplished on English i.e. the enter statements should be in English.
2) Input from the consumer is taken within the form of questions (wh - phrases like what, who, where, etc).
3) To allow smooth get admission to database.
4) Nontechnical individual also can access database without using square queries.
5) Translate the textual content into any other language.
6) Solution questions about the contents of the textual content.
7) Use of database is global-huge nowadays so NLIDB can permit easy interface for customers.
8) lets in user to interact with database gadget without knowing database schema.

©IJRASET: All Rights are Reserved

II. ANALYSIS OF NLIDB

One place of studies efforts in the query interfaces is targeted on enhancing the usability. The primary intention is to offer an excessive stage interface that can be utilized by nontechnical customers with none asked DBMS orientated expertise.

An essential place on this direction is the utility of Natural language interface for databases (NLIDB). The NLIDB manner that a consumer can use a few natural language to create question expressions and also the solution is provided within the identical language. The records of NLIDB goes again as early as 1960's. The technology of top studies interest on NLIDB became in the 1980's. In that time, the development of a website and language independent NLIDB module seemed as a realistic venture. The prototype initiatives confirmed that the building of a Natural language interface is a far greater complex venture than it become expected.

Regarding the usability of NLIDB, there may be found some tests in the literature that evaluates the efficiency of the NLI interfaces. In these assessments the NLIDB is compared with traditional interfaces like square.

The consequences display that expert users can carry out greater effectively the unique command interface (square) than the NLI interface. However, the un-skilled users could gain better outcomes with the NLI interface than with the imperative sq. interface.

Critical appraisal of other people's

The peoples who watched the running of our challenge they locate it is the amazing invention within the natural Language Processing subject. The studies paper demonstrate the idea in the back of the Processing of query while not having any Technical information.

III. COMPARATIVE STUDY

The very first attempts at NLP (natural Language Processing) database interfaces are simply as old as some other NLP research. In truth database NLP can be one of the maximum crucial successes in NLP because it began. Asking questions to databases in natural language is a totally handy and smooth approach of records get admission to. Particularly for casual users who do no longer recognize complicated database query languages consisting of square. The success in this area is in part because of the actual-global benefits that can come from database NLP structures, and partly due to the fact NLP works thoroughly in a single-database domain. Databases usually provide small sufficient domains so that ambiguity problems in Natural language can be resolved efficiently. Here are some types of database NLP structures:

Types of DB interface

1) LUNAR (Woods. 1973) involved a system that answered questions about rock samples brought back from the moon. Two databases were used, the chemical analyses and the literature references. The program used an Augmented Transition Network (ATN) parser and Woods Procedural Semantics. The system was informally demonstrated at the Second Annual Lunar Science Conference in 1971.

2) LIFER/LADDER was one of the first good database NLP systems. It was designed as a natural language interface to a database of information about US Navy ships. This system, as described in a paper by Hendrix (1978), used a semantic grammar to parse questions and query a distributed database. The LIFER/LADDER system could only support simple one-table queries or multiple table queries with easy join conditions.

IV. SYSTEM DESIGN

A. Architectural Design

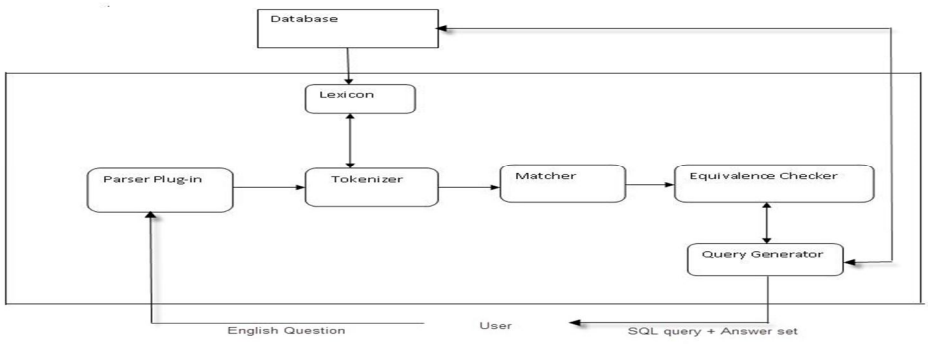


Fig. Architectural Design

B. User interface design

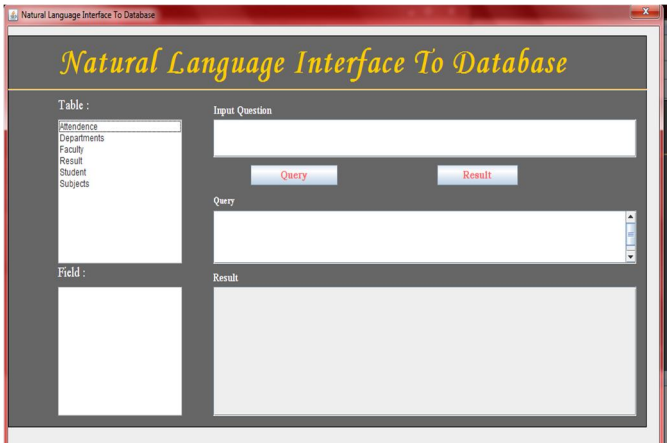


Fig. User Interface Design

C. Algorithmic Description of Each Module

1) Tokenize:-

Input:-Input Question.

Output:-Separated Tokens from input question.

a) Start (enter query)

b) Be given natural language query in String 'S'.

Three test the syntax of query (Parse the input question)

Parse (String s)

If question includes unique characters, extra spaces then remove them.

c) If question begin with 'wh' phrase or command like questions then syntax is correct.

Cut up the query in tokens 'S'

Else

File Invalid query layout message.

d) If syntax is correct then split question in tokens & take away the Syntactic markers

If tokens[S] contains Syntactic Markers then

Cast off Syntactic markers

Else

Display the Separated tokens

e) Display separated Tokens

f) Stop

2) MATCHER:

Input:-Separated Tokens

Output:-Table name and Attribute names.

a) Accept the tokens in String 'W'.

b) ApplyTokenizer (String W)

For all Tokens W[i]

Get the all synonym from the lexicon: GetTableColumnSingleRecord();

Check each element from 'W' with Synonyms.

If w[i] matches with table_name or table_name_synonym then

Found Table_name

End if

-Get all Column names of Table_name: GetColumns(Table_name)

-if W[i] matches with Colums_name or Colums_name_Synonyms then

Found Column_name
End if
-Get all values of Column_name:
GetRecords (Column_name)
-if W[i] matches with Values then
Found values
End if
End

c) Get Table name, Column names, and Values in Value_set.
d) End

3) SQL Translation:-
Input: - Table_name, Column_names, Value_set.
Output:- SQL query.
a) Accept the Table_name, Column_names, and Value_set.
b) Map the Table_name and Column_names to SQL query format: Query generator ();
-place column names and Table_name at proper position in SQL Query
-If value set found then
Attach value set to the generated query with 'Where' Clause
-End if
c) Display Generated SQL Query on User Interface.

4) Display Result:-
Input:- Table_name, Column_names, Value_set..
Output:- SQL query.
a) Get the generated SQL Query in String STR.
b) Fire the SQL query on DataBase: GetTableContents(str).
c) Get the Result in array and display in table.

V. IMPLEMENTATION METHODS

1) Lexicon
Input- Database
Output- Relation Names, Attribute Names
Lexicon is a records dictionary which consists of all of the synonyms of the database attributes. All attribute name and relation names are fetched with the aid of lexicon and stored as dictionary.
In Lexicon feature the names of all database factors (Relation and characteristic names) are extracted and cut up into man or woman phrases. Every word has set of synonyms saved on this database. Every database detail is for this reason related to a hard and fast of word steam and every word steam in flip associated with set of synonyms. Lexicon is a records dictionary which consists of all of the synonyms of the database attributes. Lexicon also consists of query codecs saved in it, which can be required whilst a natural language question is wanted to map to sq. question.

2) Tokenize
Input-English Statement
Output-Separated Tokens
Tokenizer makes use of Lexicon to map the tokens, as synonyms of token may be used by person.
Natural language sentences are given as enter to Tokenizer. Work of tokenizer is to split the tokens from the sentence. Tokenizer uses Lexicon to find the Tokens from the Natural language Sentence. The tokenizer proceeds by using steaming each word within the question and looking up in the lexicon. It additionally keep in mind the synonyms of the words in the sentence and separates the tokens.

3) Matcher

Input-Tokens

Output-Database Elements

Matcher takes tokens as enter from tokenizer and all of the tokens are mapped to the database elements. After which database factors are passed to query generator to generate question.

4) Query Generator

Input-Database Elements

Output-SQL Query

The question Generator takes the database elements decided on by using the matcher and weaves them right into a nicely-formed square question. Query generator uses database to generate square query as formats of the sq. queries are saved in database. It takes the database factors from the matcher and square question layout from the database and places all elements at appropriate places in the sq. query and generates sq. query. Generated sq. query is fired on database and result set is displayed on output display.

VI. APPLICATIONS

This system can used to retrieve the data from large databases like:

1) College Administration System.
2) Railway reservation and enquiry machine.
3) Customer care services
4) Computerized and Online Dictionaries

VII. CONCLUSION AND FUTURE WORK

To paintings with any RDBMS one should know the syntax of the instructions of that precise database software program (Microsoft square. Oracle, and many others.),

Here the Natural language processing is done on English i.e. the enter statements should be in English.

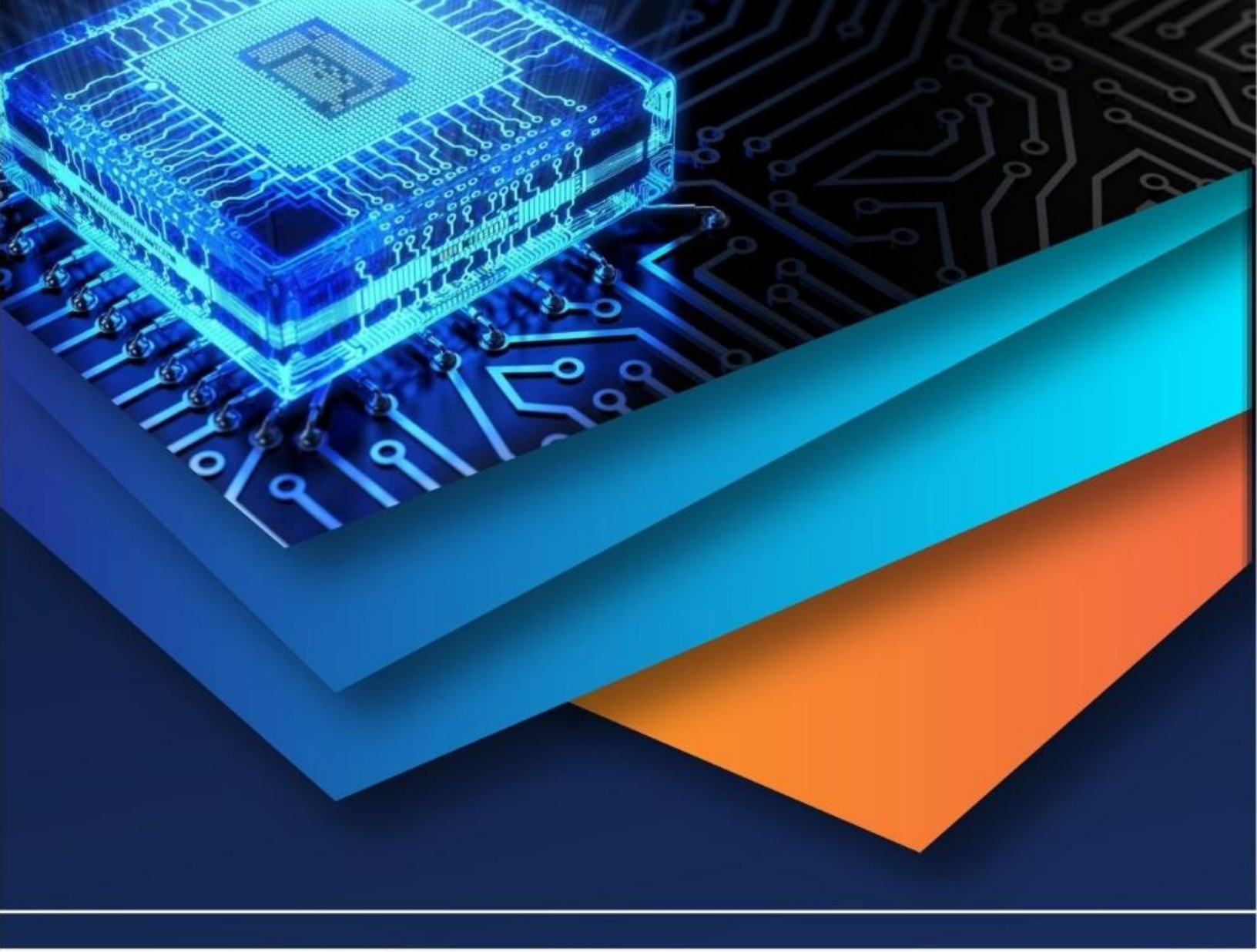
Enter from the person is taken within the form of questions (wh - phrases like what, who, wherein, and so on).

The modern implementation already has the power to music the procedure of transforming a textual query right into a formal one, which become delivered to aid better consumer interplay inside the destiny. One instance of the use of this would be to provide, on request, a justification of how the present day results are related to the query. The users who dislike black-field systems may also use this to better apprehend how the question transformation works which, in turn, may want to improve the performance of them using the system.

REFERENCES

[1] Towards a Theory of Natural Language Interfaces to Databases By Ana-Maria Popescu ,University of Washington Computer Science (2003).
[2] A Natural Language Query Interface to Structured Information By Valentin Tablan, Danica Damljanovic and Kalina Bontcheva.
[3] Intelligent Natural Language Query Interface for Temporal Databases By Ramsubramaniam P and Kannan A.
[4] J.Allen.G Ferguson ,and A.An Stent.An architecture for more realistic conversational systems .In Intelligent user Intrface,2001.
[5] I.Androutsopoulos, G.D. Ritchic , and P.Thanisch Natural Language Interface to Database –An Introduction .In Natural Language Engineering ,vol 1,part 1,pages 29-81,1995.
[6] C.Charniak. A Maximum-Entropy-Inspired Parser.In Proceeding of NAACL-2000,2000.
[7] C. Baik, H. V. Jagadish, and Y. Li, "Bridging the semantic gap with SQL query logs in natural language interfaces to databases," 35th International Conference on Data Engineering (ICDE), 2019: IEEE, pp. 374-385.
[8] P. Utama et al., "An End-to-end Neural Natural Language Interface for Databases," arXiv preprint arXiv:1804.00401, 2018.
[9] F. Li and H. Jagadish, "Constructing an interactive natural language interface for relational databases," Proceedings of the VLDB Endowment, vol. 8, no. 1, pp. 73-84, 2014.
[10] A. Neelakantan, Q. V. Le, M. Abadi, A. McCallum, and D. Amodei, "Learning a natural language interface with neural programmer," arXiv preprint arXiv:1611.08945, 2016.
[11] Y. Zhu, E. Yan, and I.-Y. Song, "A natural language interface to a graphbased bibliographic information retrieval system," Data & Knowledge Engineering, vol. 111, pp. 73-89, 2017.
[12] E. Oro and M. Ruffolo, "A Natural Language Interface for Querying RDF and Graph Databases," in Consiglio Nazionale delle Ricerche Istituto di Calcoloe Reti and Alte Prestazioni, 2015.
[13] I. Androutsopoulos, G. D. Ritchie, and P. Thanisch, "Natural language interfaces to databases–an introduction," Natural language engineering, vol. 1, no. 1, pp. 29-81, 1995.

[14] F. Brad, R. C. A. Iacob, I. A. Hosu, and T. Rebedea, "Dataset for a Neural Natural Language Interface for Databases (NNLIDB)," in Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers), 2017, pp. 906-914.
[15] H. Li and Y. Shi, "A wordnet-based natural language interface to relational databases," in 2010 The 2nd International Conference on Computer and Automation Engineering (ICCAE), 2010, pp. 514-518.
[16] K. Affolter, K. Stockinger, and A. Bernstein, "A Comparative Survey of Recent Natural Language Interfaces for Databases," arXiv preprint arXiv:1906.08990, 2019.
[17] L. Perkins, E. Redmond, and J. Wilson, Seven databases in seven weeks: a guide to modern databases and the NoSQL movement. Pragmatic Bookshelf, 2018.
[18] M. Hunger, "From Relational to Graph: A Developer's Guide," in DZone, ed, 2016.
[19] Wu, D.; Wu, C. Research on the Time-Dependent Split Delivery Green Vehicle Routing Problem for Fresh Agricultural Products with Multiple Time Windows. Agriculture 2022, 12, 793.
[20] Çinakli, M.; Merve, A.R.T.A.; Demirda ˘g, M.; Seçkin, A.Ç. A Virtual Assistant Design and Application on Industrial Database. Uluslararası Yönetim Bili¸sim Sist. Ve Bilgi. Bilimleri Derg. 2021, 5, 122–143.
[21] Chen, H.; Miao, F.; Chen, Y.; Xiong, Y.; Chen, T. A hyperspectral image classification method using multifeature vectors and optimized KELM. IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. 2021, 14, 2781–2795
[22] Ganeshkumar, D.R.; Prabhavathy, R. A Model for Analysis of Key Performance Indicators in Manufacturing Industries. IRE J. 2021, 5, 1702961.
[23] Yalçınkaya, T. New Technologies and Economic Policies in the Global System. In The Impact of Artificial Intelligence on Governance, Economics and Finance; Springer: Singapore, 2021; Volume I, pp. 139–163.
[24] Melluso, N.; Grangel-González, I.; Fantoni, G. Enhancing Industry 4.0 standards interoperability via knowledge graphs with natural language processing. Comput. Ind. 2022, 140, 103676.
[25] Ashfaq, M.; Yun, J.; Yu, S.; Loureiro, S.M.C. I, Chatbot: Modeling the determinants of users' satisfaction and continuance intention of AI-powered service agents. Telemat. Inform. 2020, 54, 101473
[26] Aviv, A.; Oshrat, Y.; Assefa, S.A.; Mustapha, T.; Borrajo, D.; Veloso, M.; Kraus, S. Advising Agent for Service-Providing Live-Chat Operators. arXiv 2021, arXiv:2105.03986.
[27] Jamwal, A.; Agrawal, R.; Sharma, M.; Giallanza, A. Industry 4.0 technologies for manufacturing sustainability: A systematic review and future research directions. Appl. Sci. 2021, 11, 5725
[28] Perazzoli, S.; de Santana Neto, J.P.; de Menezes, M.J.M.B. Systematic analysis of constellation-based techniques by using Natural Language Processing. Technol. Forecast. Soc. Chang. 2022, 179, 121674.
[29] Singhal, N. An empirical investigation of Industry 4.0 preparedness in India. Vision 2021, 25, 300–311.
[30] Zhou, K.; Liu, T.; Zhou, L. Industry 4.0: Towards future industrial opportunities and challenges. In Proceedings of the IEEE 2015 12th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD), Zhangjiajie, China, 15–17 August 2015; pp. 2147–2152.
[31] Jamwal, A.; Agrawal, R.; Sharma, M.; Giallanza, A. Industry 4.0 technologies for manufacturing sustainability: A systematic review and future research directions. Appl. Sci. 2021, 11, 5725.
[32] Xu, X.; Lu, Y.; Vogel-Heuser, B.; Wang, L. Industry 4.0 and Industry 5.0—Inception, conception and perception. J. Manuf. Syst. 2021, 61, 530–535.
[33] Acioli, C.; Scavarda, A.; Reis, A. Applying Industry 4.0 technologies in the COVID–19 sustainable chains. Int. J. Product. Perform. Manag. 2021, 70, 988–1016.
[34] Laghari, A.A.; Wu, K.; Laghari, R.A.; Ali, M.; Khan, A.A. A review and state of art of Internet of Things (IoT). Arch. Comput. Methods Eng. 2021, 29, 1395–1413.
[35] Wijethilaka, S.; Liyanage, M. Survey on network slicing for Internet of Things realization in 5G networks. IEEE Commun. Surv. Tutor. 2021, 23, 957–994
[36] Sinha, B.B.; Dhanalakshmi, R. Recent advancements and challenges of Internet of Things in smart agriculture: A survey. Future Gener. Comput. Syst. 2022, 126, 169–184.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)