A Framework for Web Based Language Independent Semantic Question Answering System

Vaibhav Agarwal[a,*], Parteek Kumar[b]
[a]*Computer Science and Engineering Department, TIET, Patiala, India*
[b]*Computer Science and Engineering Department, TIET, Patiala, India*

Abstract. Question answering systems (QAS) attempt to let you ask your question the way you'd normally ask in natural language. After doing an exhaustive survey on the existing QAS it was observed that there is no system which has all the features viz. online availability, multilinguism support, ability to extend the support to other languages without changing architecture/code base, ability to integrate other ML/NLP applications, availability of source code, and whose working of all components is known. This paper presents a framework and the live application developed (as a proof of concept) on top of this framework which supports all these features. After testing the proposed system on two standard corpora, the 'Conciseness', 'Relevance', 'Correctness', 'Precision', 'Recall', and 'F-Measure' of the developed system came out to be 89.5%, 86.4%, 100%, 86.4%, 100%, and 92.7% respectively.

Keywords: Question Answering Systems, Knowledge and data engineering tools and techniques, Natural language processing, Linguistic processing

1. Introduction

This paper presents a UNL (Universal Networking Language) based framework for language independent semantic question answering system (QAS). As a proof of concept, a QAS has been developed on top of this architecture which is web-based. The organization of this paper is structured as follows. Section 1 presents the brief idea about UNL and Section 2 presents the literature review. In order to understand the current system, the previous work done in this area is presented in Section 3. Section 4 describes the corpus used for experimentation. The architecture and phase-wise working of the proposed system has been explained in Section 5 with the help of example sentence. Section 6 describes the data structures and pseudocode of the QAS developed on proposed framework. Section 7 describes the evaluation metrics and Section 8 illustrates the methodology to calculate the metrics value. The empirical results of the proposed system are presented in Section 9. Section 10 concludes the paper and presents the future implications.

In recent few years, in the field of ML/NLP, researchers had immensely explored UNL. UNL is an intermediate language to describe, summarize, store, and represent information in a natural-language-independent format. UNL can be used in speech to speech machine translation, text simplification, summarization, machine translation, multilingual document generation, information retrieval and extraction, text mining, sentiment analysis, *etc*. UNL approach is better than other approaches because UNL represents 'what was meant' (and not 'what was said'), UNL is not bound to translation, UNL is computable, UNL is compositional, UNL representation does not depend on any implicit knowledge, *i.e.*, it is self sufficient, UNL is declarative, UNL is non-ambiguous, UNL is complete, and UNL is non-redundant. A general idea of UNL and its specifications have been provided by Uchida *et al*. (1999) [1].

Computational linguists and researchers across the globe have participated actively in UNL since the starting of the UNL programme in 1996. English, Russian, Chinese, Slovenian, Hindi, Greek, Georgian, Portuguese, Punjabi, Ukrainian, and Vietnamese *etc*.

*E-mail: vaibhavagg123@gmail.com;vaibhav.agarwal@thapar.edu.

are some of the languages which are part of the UNL programme. UNL programme is currently being managed by UNDL foundation. UNLization and NLization are the two basic processes in UNL. UNLization is the process of converting natural language text into UNL whereas NLization is the process of generating text in a natural language from UNL. UNLization is done with the help of a tool called IAN (Interactive Analyzer) while NLization is done by EUGENE (dEep-to-sUrface GENErator). Both IAN and EUGENE are online tools developed by UNDL foundation.

In UNL, three different types of semantic units are used to convey the information, namely, Universal Words (UWs), Universal Attributes, and Universal Relations. Consider UNL representation of English sentence ‘The girl throw tomatoes in the bedroom’ given in Eq. (1).

{unl}
plc(throw, bedroom.@def)
obj(throw, tomato.@pl)
agt(throw, girl.@def)
{/unl} (1)

UWs, Attributes, and Relations of UNL given in Eq. (1) are shown with the help of UNL graph in Figure 1.

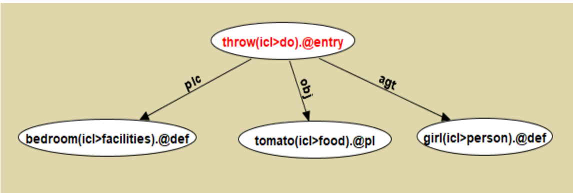

Fig. 1. UNL graph of sentence ‘The girl throw tomatoes in the kitchen’

In the given example, there are four UWs (throw, bedroom, tomato, girl), three Universal Attributes (@entry, @def, @pl), and three Universal Relations (plc, obj, agt). The relation between ‘*throw*’ (verb) and ‘*bedroom*’ (where the action took place) is specified by ‘*plc*’. The relation between ‘*tomato*’ (object) and ‘*throw*’ is specified by ‘*obj*’.

The relation between an agent ‘*girl*’ (who did work), and ‘*throw*’ is specified by ‘*agt*’. The circumstance, under which a node is being used, is represented by Universal Attributes. Universal Attributes are the annotations made to nodes. In the given example sentence shown in Eq. (2), ‘*@pl*’ indicates that UW is used as a plural. The attribute ‘*@def*’ is a kind of specifying attribute used in case of general specification (normally conveyed by determiners) which is represented by node ‘*the*’. Any additional information which cannot be expressed with the help of Universal Relations or Universal Words is expressed by Universal Attributes.

2. Literature Review

During the last couple of years, in the field of natural language processing, QA systems have been an area which had attracted researchers. This chapter focuses on the analysis and study done of various QA systems, and comparison of some important QA systems. After doing an exhaustive survey on various QA systems it has been observed that the proposed UNL based QA system will definitely be a major step in natural language processing and removing the language barrier. Table 1 and Table 2 show the analysis of the important QAS.

Table 1

QA Systems Features Comparison

QA Details / QA Research	QA System Features				
	Available Online	**Supports Multi Language**	**Can be extended to support other languages**	**Can integrate other NLP applications**	**Is Source Code Available**
Complex Sequential Question Answering (Amrita *et al*., 2018) [2]	×	×	×	×	×

		(English)			
EARL (Mohnish *et al*., 2018) [3]	×	× (English)	×	×	×
A multi-agent based Question Answering System (Abderrazzak *et al*., 2018) [4]	×	× (English)	×	×	×
Visual Question Answering using Explicit Visual Attention (Vasileios *et al*., 2018) [5]	×	× (English)	×	×	×
CQASMD (FENG *et al.*, 2018) [6]	×	× (Chinese)	×	×	×
QA4IE (Lin *et al*., 2018) [7]	×	× (English)	×	×	×
WebShodh (Chandu *et al*., 2017) [8]	✓	✓ (Hindi, English, Telugu)	×	×	×
An automatic question answering system for the Arabic Quran (Mohamed, 2017) [9]	×	× (Arabic)	×	×	×
FelisCatusZero (Kotaro *et al.*, 2017) [10]	×	× (Japanese)	×	×	×
Intelligent Question Answering System Based on Remote Service Framework (Xiaoyi *et al.*, 2017) [11]	×	× (English)	×	×	×
Question Answering System Supporting Vector Machine Method for Hadith Domain (Nabeel and Saidah, 2017) [12]	×	× (English)	×	×	×
Wikipedia Based Essay Question Answering System for University Entrance Examination (Takaaki *et al*., 2017) [13]	×	× (English)	×	×	×
A Hybrid Question Answering System based on Ontology and Topic Modeling (Kwong and Chih, 2017) [14]	×	× (English)	×	×	×
Design of Intelligent Tourism Question Answering System Based on Semantic Web (Hua and Shi-zheng, 2017) [15]	×	× (Chinese)	×	×	×
An automatic web-based question answering system for e-learning (Waheeb and Babu, 2017) [16]	×	× (English)	×	×	×
Question-answering system (Stupina *et al*., 2016) [17]	×	× (English)	×	×	×
Information Retrieval System using UNL for Multilingual Question Answering (Goel, 2016) [18]	×	× (Punjabi)	×	×	×
HPI Question Answering System (Frederik *et al*., 2016) [19]	×	× (English)	×	×	×
KSAnswer (Hyeon-gu *et al*., 2016) [20]	×	× (English)	×	×	×
WS4A (Miguel *et al*., 2016) [21]	×	× (English)	×	×	×
KBQA (Wanyun *et al*., 2016) [22]	×	× (English)	×	×	×
GENQA (Jun *et al*., 2016) [23]	×	× (Chinese)	×	×	×
A Long Short-Term Memory Model for Answer Sentence Selection in Question Answering (Wang and Nyberg, 2015) [24]	×	× (English)	×	×	×
Question Answering system using vector space model (Hartawan and Suhartono, 2015) [25]	×	× (English)	×	×	×
ISOFT (Park *et al*., 2015) [26]	×	× (English)	×	×	×
Learning Knowledge Graphs for Question Answering through Conversational Dialog (Ben *et al*., 2015) [27]	×	× (English)	×	×	×
YodaQA (Baudis, 2015) [28]	×	× (English)	×	×	✓
QCRI (Nicosia *et al*., 2015) [29]	×	✓ (Arabic, English)	×	×	×
JAIST (Quan *et al*., 2015) [30]	×	× (English)	×	×	×
CICBUAPnlp (Helena *et al*., 2015) [31]	×	× (English)	×	×	×
CASIA@V2 (Shizhu *et al*., 2014) [32]	×	×	×	×	×

		(English)			
Al-Bayan (Heba *et al*., 2014) [33]	×	× (Arabic)	×	×	×
Forst (Kotaro *et al*., 2014) [34]	×	× (Japanese)	×	×	×
Knowledge-Based Question Answering as Machine Translation (Junwei *et al*., 2014) [35]	×	× (English)	×	×	×
CMU Multiple-choice Question Answering System (Di *et al*., 2014) [36]	×	× (English)	×	×	×
A Natural Language Question Answering System in Malayalam Using Domain Dependent Document Collection as Repository (Pragisha and Reghuraj, 2014) [37]	×	× (Malayalam)	×	×	×
Watsonsim (Sean *et al*., 2014) [38]	×	× (English)	×	×	×
QASYO (Abdullah *et al*., 2011) [39]	×	× (English)	×	×	×
LOOK4 System (Avetisyan *et al*., 2010) [40]	×	× (English)	×	×	×
Interlingual information extraction as a solution for multilingual QA systems (Cardenosa *et al*., 2009) [41]	×	× (English)	×	×	×
AGRO-EXPLORER System (Surve, 2004) [42]	×	✓ (English, Hindi, Spanish, Marathi)	×	×	×
NSIR (Radev *et al*., 2002) [43]	×	× (English)	×	×	×
LASSO (Moldovan *et al*.,1999) [44]	×	× (English)	×	×	×
Lunar (Woods 1973) [45]	×	× (English)	×	×	×
Baseball (Green *et al*., 1963) [46]	×	× (English)	×	×	×
Ask Jeeves [47]	✓	✓	×	×	×

Table 2

QA Systems Key Metrics Comparison

	QA System Key Metrics				
QA Details / **QA Research**	**Corpus Used for Testing**	**Evaluation Metrics**	**Evaluation Metrics Value**	**Domain**	**Working Algorithm of all components explained**
Complex Sequential Question Answering (Amrita *et al*., 2018) [2]	Wikidata dump	Precision and Recall	Precision = 6.3% Recall = 18.4%	Open	×
EARL (Mohnish *et al*., 2018) [3]	LC-QuAD and QALD-7	Accuracy	For LC-QuAD = 0.65 For QALD-7 = 0.57	Open	×
A multi-agent based Question Answering System (Abderrazzak *et al*., 2018) [4]	Not specified	Not specified	Not specified	Not specified	×

Visual Question Answering using Explicit Visual Attention (Vasileioset *al*., 2018) [5]	Visual7W	Accuracy	0.659	Open	×
CQASMD (FENG *et al.*, 2018) [6]	Not specified	Not specified	Not specified	Closed	×
QA4IE (Lin *et al*., 2018) [7]	SQuAD	F1 Score	72.5	Open	×
An automatic question answering system for the Arabic Quran (Mohamed, 2017) [8]	Holy Quran (Surat Al-Fatiha and Al-Baqarah Chapters)	Percentage Right Answers	92.9%	Closed	×
WebShodh (Chandu *et al*., 2017) [9]	Random Questions (100 Hinglish + 100 Tenglish)	MRR	Hinglish (0.37) Tenglish (0.32)	Open	×
FelisCatusZero (Kotaro *et al.*, 2017) [10]	Few mock exams of the University of Tokyo's entrance exam	Manual	10 out of 20	Open	×
Intelligent Question Answering System Based on Remote Service Framework (Xiaoyi *et al*., 2017) [11]	Real time chat room exchanges , online discussion forums	Not specified	Not specified	Open	×
Question Answering System Supporting Vector Machine Method for Hadith Domain (Nabeel and Saidah, 2017) [12]	Hadith corpus (Pray and Fasting subjects)	F-Measure	80%	Closed	×
Wikipedia Based Essay Question Answering System for University Entrance Examination (Takaaki *et al*., 2017) [13]	QA task of NTCIR (NII Testbeds and Community for Information access Research) QA Lab 3	ROUGE-1 Mean	0.584	Open	×
A Hybrid Question Answering System based on Ontology and Topic Modeling (Kwong and Chih, 2017) [14]	80 questions with answers (20 questions per categories) from the textbooks as the gold standard	MRR	38.73%	Open	×
Design of Intelligent Tourism Question Answering System Based on Semantic Web (Hua and Shi-zheng, 2017) [15]	2000 standard question on Chongqing tourism attractions, tourism traffic and tourism catering	Precision	86%	Closed	×
An automatic web-based question answering system for e-learning (Waheeb and Babu, 2017) [16]	World Wide Web	F-Measure	80%	Open	×
Question-answering system (Stupina *et al*., 2016) [17]	Following 3 books: Life Science concepts for middle school, Biology, Earth Science Concepts For Middle School.	Accuracy	53.6%	Open	✓
Information Retrieval System using UNL for Multilingual Question Answering (Goel, 2016) [18]	UNL-EOLSS	Accuracy	81.5%	Open	✓
HPI Question Answering System (Frederik *et al*., 2016) [19]	Document provided by BioASQ 2016	Mean Average Precision	0.434	Closed	×
KSAnswer (Hyeon-gu *et al*., 2016) [20]	Document provided by BioASQ 2016	Mean Average Precision	0.3752	Closed	×
WS4A (Miguel *et al*., 2016) [21]	Document provided by BioASQ 2016	F-Measure	0.24	Closed	×
KBQA (Wanyun *et al*., 2016) [22]	QALD-5	Precision and Recall	Precision: 0.86 Recall: 0.50	Open	✓
GENQA (Jun *et al*., 2016) [23]	World Wide Web	Accuracy	52%	Open	×
A Long Short-Term Memory Model for Answer Sentence Selection in Question Answering (Wang and Nyberg, 2015) [24]	Text REtrieval Conference (TREC) QA track (8-13) data	MRR	0.7913	Open	×
Question Answering system using vector space model (Hartawan and Suhartono, 2015) [25]	Data comes from 2 ministers from Indonesia; they are Minister of Education & Culture and Minister of Tourism & Creative Economy Culture.	F-Measure	0.580	Open	×
ISOFT (Park *et al*., 2015) [26]	QALD-5 test dataset	F-Measure	0.33	Open	×
Learning Knowledge Graphs for Question Answering through Conver-	Identity, WordNet, PPDB, KNOWBOT, LEAVE-ONE-	Percentage Correct	43.7%	Open	×

sational Dialog (Ben *et al*., 2015) [27]	OUT				
YodaQA (Baudis, 2015) [28]	Public question answering benchmark from TREC 2001 and 2002, QA tracks with regular expression answer patterns and extended by questions asked to a YodaQA predecessor by internet users via an IRC interface.	Recall	79.3%	Open	×
QCRI (Nicosia *et al*., 2015) [29]	SemEval-2015 Task 3 "Answer Selection in Community Question Answering"	F-Measure	English Subtask A: 53.74 (English) English Subtask B: 53.60 (English) English Subtask A: 53.74 (Arabic) English Subtask B: 53.60 (Arabic)	Open	×
JAIST (Quan *et al*., 2015) [30]	SemEval-2015 Task 3 "Answer Selection in Community Question Answering"	F-Measure	English Subtask A : 57.18	Open	×
CICBUAPnlp (Helena *et al*., 2015) [31]	SemEval-2015 Task 3 "Answer Selection in Community Question Answering"	F-Measure	English Subtask B : 38.8	Open	×
CASIA@V2 (Shizhu *et al*., 2014) [32]	QALD-4 dataset	F-Measure	0.36	Open	×
Al-Bayan (Heba *et al*., 2014) [33]	Holy Quran	Accuracy	85%	Closed	×
Forst (Kotaro *et al*., 2014) [34]	Japanese textbooks and Wikipedia	ROUGE Scores	ROUGE-1 Score: 0.125 ROUGE-2 Score: 0.062 ROUGE-L Score: 0.097	Open	×
Knowledge-Based Question Answering as Machine Translation (Junwei *et al*., 2014) [35]	WEBQUESTIONS CORPUS (3,778 questions)	Accuracy and Precision	32.5% (Accuracy) 73.2% (Precision)	Open	✓
CMU Multiple-choice Question Answering System (Di *et al*., 2014) [36]	NTCIR-11 QA Lab evaluations	Accuracy	51.6%	Open	×
A Natural Language Question Answering System in Malayalam Using Domain Dependent Document Collection as Repository (Pragisha and Reghuraj, 2014) [37]	The system was tested with 100 different questions included in the four classes of question words. Ten documents in the domain of Indian history and geographical features of India were used for the testing.	Accuracy	81%	Closed	✓
Watsonsim (Sean *et al*., 2014) [38]	Jeopardy	Accuracy	18%	Open	×
QASYO (Abdullah *et al*., 2011) [39]	YAGO ontology version 2008-W40-2 (100 questions)	Accuracy	91%	Open	×
LOOK4 System (Avetisyan *et al*., 2010) [40]	World Wide Web	Accuracy	80.6%	Open	×
Interlingual information extraction as a solution for multilingual QA systems (Cardenosa *et al*., 2009) [41]	UNL-EOLSS	Accuracy	82.6%	Open	×
AGRO-EXPLORER System (Surve, 2004) [42]	Agro Explorer Corpus	Accuracy	73.52%	Closed	×

NSIR (Radev *et al.*, 2002) [43]	TREC-8 (200 questions)	MRR	0.974	Open	×
LASSO (Moldovan *et al.*,1999) [44]	TREC-8 (200 questions)	Accuracy	55.5% (short answers) 64.5% (long answers)	Open	×
Lunar (Woods, 1973) [45]	Chemical analysis data on lunar soil and rock composition	Accuracy	78.2%	Restricted	×
Baseball (Green *et al.*, 1963) [46]	Baseball games played in the American league over one season.	Accuracy	84.38%	Restricted	×
Ask Jeeves [47]	World Wide Web	Not specified	Not specified	Open	×

3. Previous Work Done

Before presenting the architecture and details of the developed system, it is important to have the knowledge of the work which had been done earlier in UNL (which has been extended and on top of which the proposed system has been built).

- UNLization/IAN module has been developed by
- Agarwal and Kumar (2018) for Punjabi natural language [48].
- A public platform for developing language-independent applications has been developed and tested (Agarwal and Kumar (2016)) [49]. It calls IAN and EUGENE module of the source and target natural language for UNLization and NLization respectively.
- NLization/EUGENE modules have been created by Singh (2013) and Verma (2013) for Punjabi natural language [50][51].

4. Corpus Details Used For Experimentation

The proposed system has been implemented and tested on two corpora, *i.e.*, UNL-EOLSS and Agro-Explorer. The UNL-EOLSS corpus has been taken from World's largest online publication repository "The Encyclopedia of Life Support Systems (EOLSS)" that is available at *http://www.eolss.net*. The UNDL foundation, under the project UNL-EOLSS, created the content of 30 articles of the Encyclopedia of Water, one of the many encyclopedias of EOLSS[1]. The UNDL foundation has made available UNL-EOLSS corpus which has 25 articles and 12,917 sentences[2]. To test the proposed system, questions are framed on these 25 articles.

Agro-Explorer is an agricultural domain corpus developed by CFILT, IIT Bombay[3] and has been splitted into 7 sections. It comprises of total 240 complex sentences. Questions are framed on these 7 sections so as to test the proposed system on another corpus.

5. Architecture of the Proposed System

The architecture of the proposed system is shown in Figure 2.

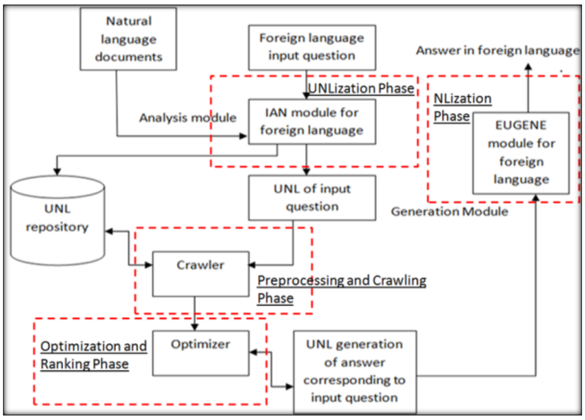

Fig. 2. The architecture of the proposed system

The architecture/working of the proposed system has been divided into UNLization phase, Preprocessing and Crawling phase, Optimizing and Ranking phase, and NLization phase. Each of the architecture modules and system's phase wise working is explained in following subsection with the help of an example sentence.

[1]http://www.unlweb.net/wiki/EOLSS
[2]http://www.undlfoundation.org/eolss/corpus/UNL.txt
[3]http://www.cfilt.iitb.ac.in

5.1. Architecture Components and their Phase wise Working with the help of an Example Sentence

Here, every architecture component along with its real-time implementation is explained phase wise with the help of an example sentence.

i. Analysis Module (UNLization Phase): The purpose of the analysis module is to convert natural language sentence to UNL. The Analysis Module calls IAN of the source natural language. In order to UNLize any given corpus or question asked by the user, IAN uses the natural language dictionary, normalization rules (NRules), transformation rules (TRules) and disambiguation rules (DRules) of the source document's natural language. These resources are created according to UNL specifications [19]. The web-based public platform developed by Agarwal and Kumar (2017) has been updated. UNL Crawler and Optimizer has been developed and integrated with this platform. Users can write their queries in their natural language and find the answer with the help of this interface as shown in Figure 3.

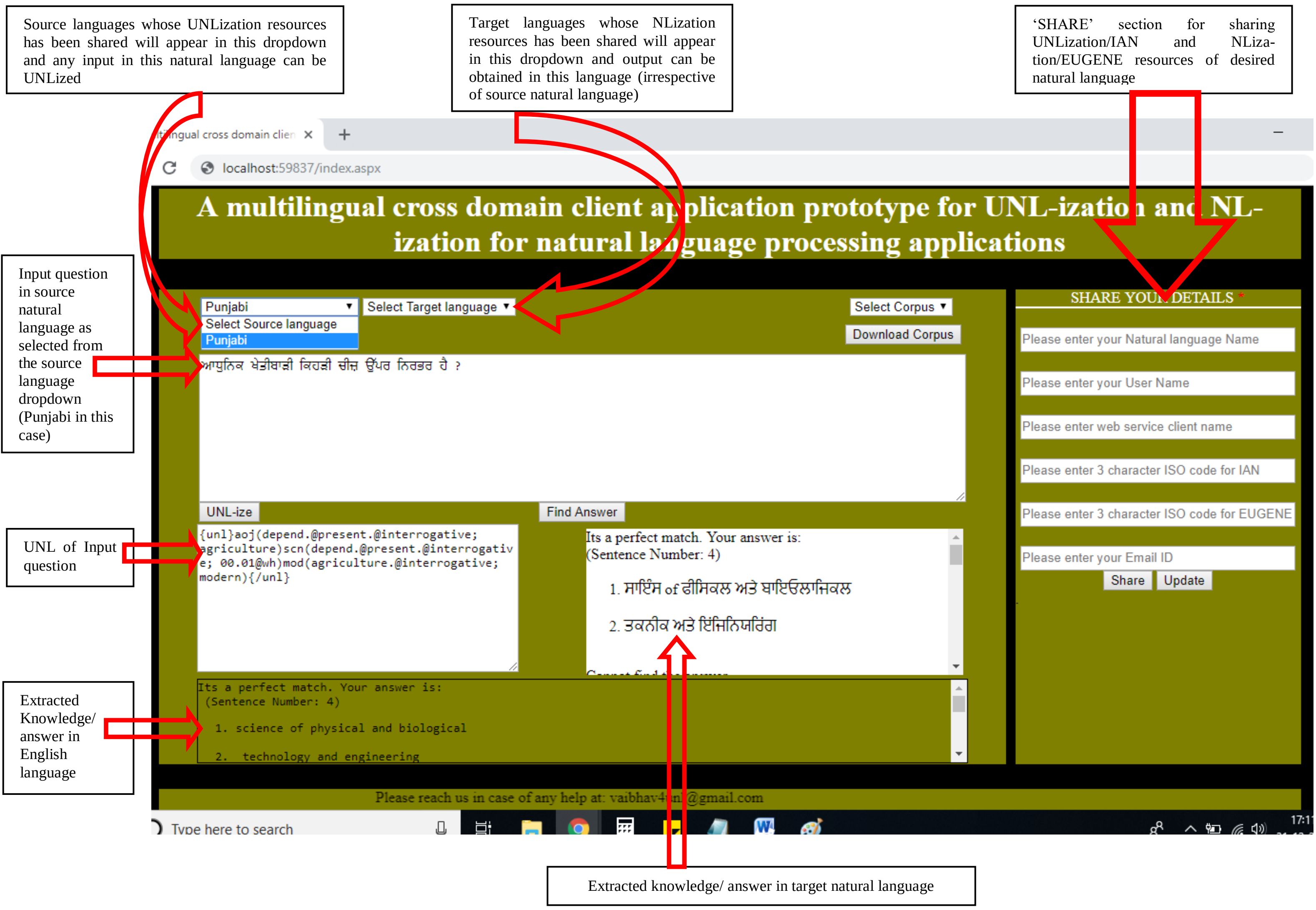

Fig. 3. The proposed web-based system interface

Let’s say the user selects Agro-Explorer corpus from the drop down and selects ‘Punjabi’ as the natural language from the drop-down options. Consider an example sentence shown in Eq. (2) taken from Agro-Explorer.

ਆਧੁਨਿਕ ਖੇਤੀਬਾੜੀ ਬਹੁਤ ਹੱਦ ਤੱਕ ਇੰਜਿਨਿਯਰਿੰਗ ਅਤੇ ਤਕਨਾਲੋਜੀ ਅਤੇ ਬਾਇਯੋਲਾਜਿਕਲ ਤੇ ਫਿਜ਼ਿਕਲ ਸਾਇੰਸ ਉੱਪਰ ਨਿਰਭਰ ਹੈ

Ādhunika khētībāṛī bahuta hada taka iñjiniyariṅga atē takanālōjī atē bā'iyōlājikala tē phizikala sā'isa upara nirabhara hai

Modern agriculture depends heavily on engineering and technology and on the biological and physical science (2)

The UNL of this sentence and all other sentences of Agro-Explorer is present in AgroExplorer.txt file which is the part of this complete developed system solution and forms the ‘UNL Repository’. Let’s assume that the user asks the following question through the interface as shown in Figure 3.

ਆਧੁਨਿਕ ਖੇਤੀਬਾੜੀ ਕਿਹੜੀ ਚੀਜ਼ ਉੱਪਰ ਨਿਰਭਰ ਹੈ?

Ādhunika khētībāṛī kihaṛī cīza upara nirabhara hai?

Modern agriculture depends on what? (3)

When the user clicks on the ‘Find Answer’ button, the Analysis module for ‘Punjabi’ natural language is called and the UNL of the input question is obtained as shown in Figure 3. UNL of the question asked in Eq. (3) is given in Eq. (4).

{unl}
aoj(depend@present.@pred.@interrogative, agriculture)
scn(depend.@present.@pred.@interrogative, 00.01@wh)
mod(agriculture.@interrogative,modern)
{/unl} (4)

ii. UNL Crawler (Preprocessing and Crawling Phase): UNL crawler is responsible for searching the UNL repository depending on the question’s UNL generated in the previous step. This module is a part of the web-based system interface (shown in Figure 3). In UNL Crawler preprocessing happens, which preprocesses the input UNL and the target UNL repository and removes attributes and scope values which are not required for finding the answer. The UNL of the input question given in Eq. (4), after preprocessing is shown in Eq. (5).

{unl}
aoj(depend, agriculture)
scn(depend, 00.01@wh)
mod(agriculture, modern)
{/unl} (5)

Similarly, after preprocessing, the UNL of the example sentence shown in Eq. (2) is given in Eq. (6).

{unl}
aoj(depend, agriculture)
scn(depend, :03)
man(depend, heavily)
mod:03(science, :02)
and:02(physical, biological)
and:03(technology, engineering)
mod(agriculture, modern)
{/unl} (6)

Once preprocessing is completed, UNL Crawler parses answer UNL and question UNL and attempts to find the answer. For the current example sentence, the Crawler finds the answer as “technology and engineering”, “science of physical and biological” as shown in Figure 3.

iii. Optimizer (Optimizing and Ranking Phase): Optimizer analyses and tells if the answer provided by ‘UNL Crawler’ is a full match/partial match, if the question asked is correct, if the answer can be found. Here, the system is in its Optimizing phase.

As shown in Figure 3, it’s the Optimizer which tells that “It’s a perfect match. Your answer is”. After Optimizing phase, in the Ranking phase Optimizer might give the following rank to the answer found by the Crawler:

a. It’s a perfect match. Your answer is:
b. It’s ALMOST a perfect match. We cannot find some information in the Database. The answer is:
c. It’s a partial match. The most probable answer is:
d. Your question is correct but we are sorry because our database does not contain sufficient information to answer this.
e. Cannot find the answer

iv. Generation Module (NLization Phase): Generation module uses EUGENE tool. Similar to Analysis Module in order to NLize the output UNL by EUGENE, TRules, DRules, and generation dictionary is created by the computational linguists of the respective natural language and stored at UNL web. These resources are created according to UNL specifications [1]. This generation module converts the output of Optimizer to the target natural language.

UNL Crawler and Optimizer work on UNL (which is natural language independent). Based on the user selection of the question’s natural language its Analysis module is called and then question’s UNL is given to UNL Crawler. Subsequently, answer keyword generated by Optimizer is given to the

Generation module for converting it to the desired target natural language by calling its EUGENE module. In the current example, the Generation Module of the 'Punjabi' natural language is called and the output given by the UNL Crawler is NLized. If EUGENE is not able to NLize the keywords, then that keyword is returned as it is. Figure 3 shows the final answer in the desired target language (Punjabi).

In the proposed system, the questions have been categorized into three different types:

i. ***Missing type question***: These are the questions in which a node in UNL graph has an argument '00.01@wh'. For example: Consider cleaned UNL given in Eq. (7) of a question "On what does modern agriculture depends".

{unl}
mod(agriculture, modern)
aoj(depend, agriculture)
scn(depend, 00.01@wh)
{/unl} (7)

ii. ***Polar (Yes/No type) question***: These are the questions in which a node in UNL graph has an argument 'xyz@YesNo'. Here 'xyz' means any combination of characters and 'YesNo' is the attribute which is given during UNLization of the input natural language question sentence. For example: Consider cleaned UNL given in Eq. (8) of a question "Does modern agriculture depends on engineering".

{unl}
mod(agriculture, modern)
aoj(depend, agriculture)
scn(depend, engineering@YesNo)
{/unl} (8)

iii. ***Non- missing type question***: These are the questions in which a node in UNL graph has an argument 'xyz@wh'. Here 'xyz' means any combination of characters (except '00.01' which is used for missing type questions). For example: Consider cleaned UNL given in Eq. (9) of a question "Does modern agriculture depends on engineering".

{unl}
aoj(include, agriculture@wh)
{/unl} (9)

6. Data Structures and Pseudocode

During the preprocessing phase in which answer's and question's UNL is parsed and cleaned up, the base data structures *i.e.*, JSON objects are created. For storing Answer UNL information 'answerObject' is formed as shown in Figure 4. Similarly for storing Question UNL information 'questionObject' is formed as shown in Figure 5. These objects have following keys:

a. 'firstArgument' which stores first argument of UNL relation.
b. 'secondArgument' which stores second argument of UNL relation.
c. 'relationName' which stores name of UNL relation.
d. 'otherInfo' which holds any other information. In the proposed system it has default value 'no'. This key has been added just to make sure that in future if anything needs to be updated or any other information/ functionality is required, and then this key can be used.

Fig. 4. Web browser's (google chrome) debugger snapshot showing data structure for storing answer UNL

Fig. 5. Web browser's (google chrome) debugger snapshot showing data structure for storing question UNL

For storing information related to hyper nodes/ scope nodes, 'hyperNodes' object is created. For the given example sentence shown in Eq. (2) its 'hyperNodes' object is as shown in Figure 6.

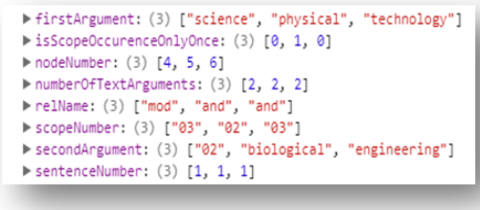

Fig. 6: Web browser's (google chrome) debugger snapshot showing data structure for storing scope/ hyper UNL graph information

These above mentioned data structures are used by UNL Crawler to crawl and parse the question and answer. Following is the pseudocode is used by UNL crawler for crawling.

PSEUDOCODE 1: Crawling

Input: Question's UNL, Answer's UNL
Output: Answer in English

1. Scan question's UNL to check if it is a missing type question, polar question or non-missing type question.
2. Call the procedure 'findAnswer' to get the best possible answer based on the type of question.

The pseudocode for identifying question type *i.e.*, if it is a missing type question, Polar (Yes/No type) question, or non-missing type question is explained as follows.

PSEUDOCODE 2: FindQuestionType

Input: Question's UNL

1. If question's UNL has a node with an argument like '00.01@wh' then it's a missing type question.
2. If question's UNL has a node with an argument like 'xyz@YesNo' then it's a polar question.
3. If question's UNL has a node with an argument like 'xyz@wh' then it's a non-missing type question.

The pseudocode for finding answer is explained as follows.

PSEUDOCODE 3: FindAnswer

Input: Question's UNL, Answer's UNL
Output: Answer in English

1. If the question is a missing type question then
    a. Store name of the relation in which question argument (00.01@wh) is present in a variable as answer relation name.
    b. Store the sibling argument of the node in which question argument (00.01@wh) is present in a variable as an argument to match.
    c. If 00.01@wh is the second argument then do the following:
        i. Crawl the answer UNL and look for a node whose relation name is same as that of answer relation name and whose first argument is same as an argument to match.
        ii. Case 1: If such a node is found and the second argument is not a scope then its second argument is the required answer and exit.
        Case 2: Else if such node is found and the second argument is a scope, then resolve that scope using 'hyperNodes' object and exit.
        Case 3: Else if any other nodes are found whose first argument is same as an argument to match, then their second arguments could be a probable answers (later decided by Optimizer) so store them and exit.
        Case 4: Else if no such node is found, then save answer as blank or empty (i.e., answer = "") and exit.
    d. If 00.01@wh is the first argument then follow step '1-c' with the only difference that instead of

the first argument use second argument and vice-versa and exit.

2. If the question is a polar question then
    a. Store name of the relation in which question argument (xyz@YesNo) is present in a variable as answer relation name. Call this node as candidate node.
    b. Store the sibling argument of the node in which question argument (xyz@YesNo) is present in a variable as an argument to match.
    c. Crawl the answer UNL and check if candidate node is present or not. If the candidate node is present then the answer is 'Yes' and exit.
    d. If xyz@YesNo is the second argument then do the following:
        i. Crawl the answer UNL and look for a node whose relation name is same as that of answer relation name and whose first argument is same as an argument to match.
        ii. If such node is found and the second argument is a scope, then resolve that scope using 'hyperNodes' object and check if 'xyz' is in any of the substrings of that resolved hyperNode.
        iii. If 'xyz' is in any of the substrings of that resolved hyperNode then set answer as 'Yes'
        Else set answer as 'No' and exit.
    e. If xyz@YesNo is the first argument then follow step '2-d' with the only difference that instead of the first argument use second argument and vice-versa.
3. If the question is a non-missing type question then
    a. Store name of the relation in which question argument (xyz@wh) is present in a variable as answer relation name.
    b. Store the sibling argument of the node in which question argument (xyz@wh) is present in a variable as an argument to match.
    c. If xyz@wh is the second argument then do the following:
        i. Crawl the answer UNL and look for all nodes whose first argument is same as an argument to match, and the second argument is not 'xyz'.
        ii. For all such nodes found if the second argument is not a scope then save their second arguments as the expected required answers and exit.
        Else if the second argument is a scope, then resolve that scope using 'hyperNodes' object and exit.
        Else if no such nodes are found, then save answer as blank or empty (i.e., answer = "") and exit.
    d. If xyz@YesNo is the first argument then follow step '3-c' with the only difference that instead of the first argument use second argument and vice-versa.

Once the answers are given by the Crawler, Optimizer tries to tell the user about the accuracy of the answer by assigning a rank to it. The ranking is only given to missing type questions. Polar questions and non-missing type questions does not require ranking. Pseudocode for ranking is explained below.

PSEUDOCODE 4: Optimization_Ranking

Input: Question's UNL, Answer's UNL, List of candidate answers given by UNL Crawler

Output: Answers with ranking and relevance in the proper format

1. If the question is a missing type question then
    a. Parse the question UNL and answer UNL.
    b. If all the information in question UNL is present in answer UNL and answer given by Crawler is not blank, then optimizer says that 'it's a perfect match'.
    c. If all the information in question UNL is present in answer UNL but the answer given by Crawler is blank, then optimizer reports that 'Your question is correct but we are sorry because our database does not contain sufficient information to answer this'.
    d. If not all the information in question UNL is present in answer UNL and answer is not blank, then optimizer reports that 'It's ALMOST a perfect match. We cannot find some information in Database'.

e. If all or some of the information in question UNL is present in answer UNL and If the answer found by Crawler is as per Step 1-c-ii-Case 3 in PSEUDOCODE 3 (FindAnswer), then optimizer recognizes that it's not a perfect match and says that 'It's a partial match. The most probable answer is'.

7. Metrics Used For Evaluation

The following important metrics has been used for evaluating the proposed system.

- **Conciseness:** It means that the answer should not contain irrelevant or extraneous information.
- **Relevance**: It specifies that the answer should be a response to the question.
- **Correctness:** It specifies that the answer should be factually correct.

Apart from these metrics, **Precision** and **Recall** have also been long used for evaluating QA systems. Precision is the measure of accuracy whereas Recall is the measure of its exhaustivity. Precision means what percentage of answers given by the system is relevant and is calculated by the formula given in Eq. (10). Recall means what percentage of relevant answers is returned by the system and is calculated by the formula given in Eq. (11).

$$Precision = \frac{Number \quad of \quad relevant \quad retrieved}{Total \quad retrieved} \qquad (10)$$

$$Recall = \frac{Number \quad of \quad relevant \quad retrieved}{Total \quad relevant \quad items} \qquad (11)$$

F-Measure or F1-Score is the harmonic mean of Precision and Recall and is calculated by the formula given in Eq. (12).

$$F - Measure = \frac{2 \times Precision \times Recall}{Precision + Recall} \qquad (12)$$

8. Methodology To Calculate Metrics Value

The questions that are framed out of the two corpora are assumed to have only one correct answer which can be present in any sentence of these corpora. Consider scenario in which the user asks a question to the system and the system responded with 3 answers out of which only one answer is right. In this case, the **Relevance** of this question would be 1/3 = 33.3% because 1 out of 3 answers is relevant. The **Precision** of this will be same as Relevance as out of 3 answers 1 answer is relevant. This is true for every case.

The Relevance/Precision of the complete system would be the weighted average of Relevance/Precision of all the 400 questions asked with weights equal to the total number of questions per question type.

Conciseness is calculated by the formula given in Eq. (13).

$$Conciseness = \frac{Total \quad concise \quad answers \quad obtained}{Total \quad questions \quad asked} \times 100 \qquad (13)$$

As described earlier, an answer is said to be concise if it does not have any irrelevant or extraneous information. An answer which is not concise will have Relevancy/Precision equal to 0.

Correctness is calculated by the formula given in Eq. (14).

$$Correctness = \frac{Total \ factually \ correct \ answers \ obtained}{Total \ questions \ asked} \times 100 \qquad (14)$$

Since all the answers to the questions have been extracted from the UNL repository (which is assumed to have correct UNL) and no manipulation is done by the developed system, therefore all answers are factually correct and hence 'Correctness' is 100%. Since the proposed system works on UNL of input and UNL of the corpus, therefore the Recall of the proposed system is 1 because, by design, UNL Crawler will definitely find an answer if it is present in the corpus. However, if the UNL of question or corpus is wrong, then it will reduce the Recall. Since for the polar questions, answers contain only 'Yes' or 'No', therefore Conciseness and Relevance/Precision is not applicable to this and is evaluated as 1 always.

9. Empirical Results

For testing the proposed system, 400 questions are framed and asked through the developed interface as illustrated in Figure 3. All the sentences in the UNL-EOLSS Corpus and Agro-Explorer Corpus are converted to UNL and saved as UNLEolss.txt, and AgroExplorer.txt files respectively as a part of the developed solution. Out of these 400 questions, the total number of missing type questions are 300, Polar questions are 70 and non-missing type questions are 30. Table 3 shows the count of types of questions.

Table 3

Types of Questions

Sr No.	Question Type	Count	Category
1.	Polar (Yes-No)	70	Polar question
2	What	60	Missing type question
3.	Who	50	
4.	Where	40	
5.	When	40	
6.	Why	40	
7.	Which	40	
8.	How	30	
9.	What	30	Non-Missing type question
Total	-	**400**	

After testing the proposed system on 400 questions, it has been observed that its 'Conciseness' is 89.5% as out of 400 questions, answers to 42 questions had extra information. The value of Relevance/Precision of these 42 questions will be 0. Apart from these 42 questions, there were 17 more questions, whose responses were in the form of multiple answers.

The Table 4 shows the question wise values of Conciseness, Relevance/Precision, Recall and F-Measure of the developed system and has been calculated with the help of formula given in Eqs. (10), (11), (12), and (13).

Table 4

Conciseness, Relevance/ Precision, Recall and F-Measure of the Developed System

Sr No.	Question Type	Count	Conciseness	Relevance/ Precision	Recall	F-Measure
1.	What	90	0.84	0.81	1	0.895
2.	Polar (Yes-No)	70	1.0	1.0	1	1
3.	Who	50	0.86	0.82	1	0.901
4.	Where	40	0.90	0.86	1	0.924
5.	When	40	0.95	0.93	1	0.963
6.	Why	40	0.87	0.85	1	0.918
7.	Which	40	0.85	0.80	1	0.888
8.	How	30	0.86	0.81	1	0.895
Total	-	**400**	**0.895**	**0.864**	**1**	**0.927**

Figure 7 shows the values of Conciseness, Correctness, Relevance/ Precision, Recall, and F-Measure of the developed system.

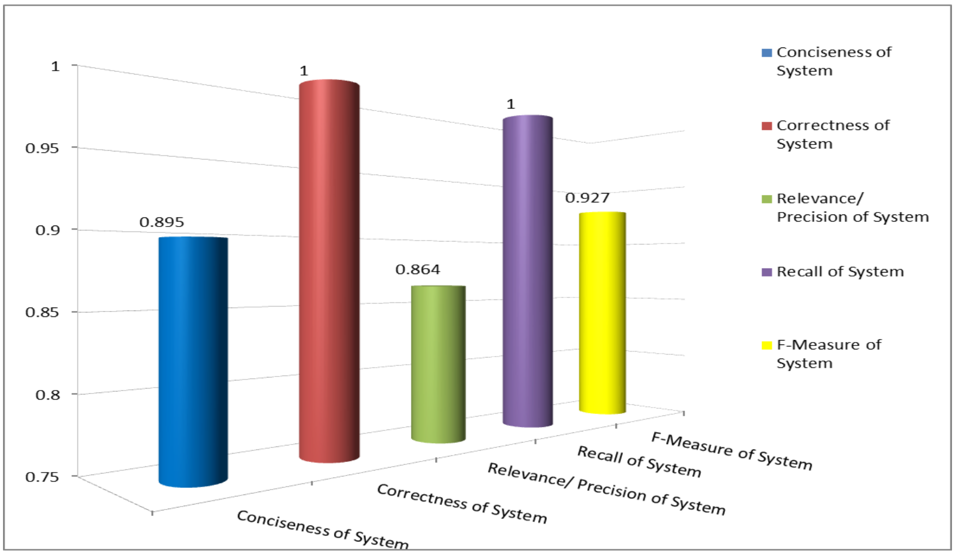

Fig. 7. Measures of Conciseness, Correctness, Relevance/ Precision, Recall, and F-Measure

10. Conclusions and Future Scope

This paper presents a UNL (Universal Networking Language) based framework for language independent semantic question answering system (QAS). As a proof of concept, a QAS has been developed on top of this architecture which is web-based. The proposed system has given very promising results after testing on UNL-EOLSS and Agro-Explorer corpora.

The ‘Conciseness’, ‘Relevance’, ‘Precision’, and ‘F-Measure’ metric values of the proposed system can be improved. IAN and EUGENE modules can be enriched for Punjabi natural language so that their F-Measure can be increased.

Since a public platform for developing language-independent applications has been developed and tested, therefore other NLP applications (UNL based) like sentiment analysis, text summarization, machine translation *etc.* can be developed and integrated with this. The developed system can be extended to support the feature to upload the UNL corpus by the user so that questions can be asked by the worldwide audience.

References

[1] H. Uchida, M. Zhu, T. D. Senta, “The UNL, A gift for a millennium,” *Institute of Advanced Studies, The United Nations University,* Tokyo, Japan, pp. 1-62, 1999.

[2] A. Saha, V. Pahuja, M. M. Khapra, K. Sankaranarayanan, and S. Chandar, “Complex Sequential Question Answering: Towards Learning to Converse Over Linked Question Answer Pairs with a Knowledge Graph,” *arXiv preprint arXiv:1801.10314*, pp. 1-9, 2018.

[3] M. Dubey, D. Banerjee, D. Chaudhuri, and J. Lehmann, “EARL: Joint Entity and Relation Linking for Question Answering over Knowledge Graphs,” *arXiv preprint arXiv:1801.03825*, pp. 1-16,2018.

[4] A. Samadi, E. F. Hanaa, M. Qbadou, M. Youssfi and F. Akef, "A syntactic and semantic multi-agent based question answering system for collaborative e-learning," *4th International Conference on Optimization and Applications (ICOA)*, pp. 1-4, 2018.

[5] L. Vasileios, N. Passalis, and A. Tefas, “Visual Question Answering using Explicit Visual Attention,” *International Symposium on Circuits and Systems (ISCAS)*, pp. 1-5, 2018.

[6] G. Feng, Z. Du, and X. Wu, “A Chinese Question Answering System in Medical Domain,” *Journal of Shanghai Jiaotong University (Science)*, 23(5), pp. 678-683, 2018.

[7] L. Qiu, H. Zhou, Y. Qu, W. Zhang, S. Li, S. Rong, D. Ru, L. Qian, K. Tu, and Y. Yu, “QA4IE: A Question Answering based Framework for Information Extraction,” *arXiv preprint arXiv:1804.03396*, pp. 1-16, 2018.

[8] K. R. Chandu, M. Chinnakotla, A. W. Black, and M. Shrivastava, “WebShodh: A Code Mixed Factoid Question Answering System for Web,” *In International Conference of the Cross-Language Evaluation Forum for European Languages*, pp. 104-111, 2017.

[9] M.A.H. Adany, “An automatic question answering system for the Arabic Quran,” *Doctoral dissertation*, Sudan University of Science and Technology, 2017.

[10] K. Sakamoto, T. Matsumoto, M. Ishioroshi, H. Shibuki, T. Mori, N. Kando, T. Mitamura, “FelisCatusZero: A world history essay question answering system for the University of Tokyo’s entranceexam,” *Open Knowledge Base and Question Answering Workshop at SIGIR*, Tokyo, Japan, pp. 1-2, 2017.

[11] Y. Xiaoyi, J. Xie, and Y. Liu, “Implementation of the Intelligent Question Answering System Based on Remote Service Framework,” *DEStech Transactions on Social Science, Education and Human Science, (icsste)*, pp. 760-766, 2017.

[12] N. Nabeel, and S. Saidah, “Question Answering System Supporting Vector Machine Method For Hadith Domain,” *Journal of Theoretical & Applied Information Technology*, 95(7), pp. 1510-1524, 2017.

[13] M. Takaaki, C. Francesco, B. Fadi, C. Evan, C. Cheng-Ta, X. Keyang, and M. Teruko, “Wikipedia Based Essay Question Answering System for University Entrance Examination,” *In Proceedings of Open Knowledge Base and Question Answering Workshop at SIGIR*, pp. 1-9, 2017.

[14] K.S. Fong, and C. H. Bong, “A Hybrid Question Answering System based on Ontology and Topic Modeling,” *Journal of Telecommunication, Electronic and Computer Engineering (JTEC*), 9(2-10), pp. 151-158, 2017.

[15] H. U. Hua, and S. Z. ZHU, “Design of Intelligent Tourism Question Answering System Based on Semantic Web,” *DEStech Transa ctions on Economics, Business and Management*, pp. 20-25, 2017.

[16] A. Waheeb, and A. Babu, “An automatic web-based question answering system for e-learning,” *Information Technology and Learning Tools*, 58(2), pp. 1-10, 2017.

[17] A Stupina, A & A Zhukov, E & N Ezhemanskaya, S & V Karaseva, M & N Korpacheva, L. (2016). Question-answering system. IOP Conference Series: Materials Science and Engineering. 155. 012024. 10.1088/1757-899X/155/1/012024.

[18] Goel, K. 2016. Information Retrieval System using UNL for Multilingual Question Answering, M.E dissertation, Thapar University, Patiala.

[19] Frederik, S., Schüler, R., Draeger, T., Dummer, D., Ernst, A., Flemming, P., ... & Neves, M. (2016). Hpi question answering system in bioasq 2016. In *Proceedings of the Fourth BioASQ workshop* (pp. 38-44).

[20] Hyeon-gu, L., Kim, M., Kim, H., Kim, J., Kwon, S., Seo, J., & Choi, J. K. (2016). KSAnswer: question-answering system of Kangwon national university and Sogang university in the 2016 BioASQ challenge. In *Proceedings of the Fourth BioASQ workshop* (pp. 45-49).

[21] Miguel, R., Falé, M., & Couto, F. (2016). WS4A: a biomedical question and answering system based on public web services and ontologies. *arXiv preprint arXiv:1609.08492*.

[22] Wanyun, C., Yanghua, X., Haxiun, W., Yangqiu, S., Seung-won, H., & Wei, W. (2017). KBQA: learning question answering over QA corpora and knowledge bases. *Proceedings of the VLDB Endowment*, *10*(5), 565-576.

[23] Jun Yin, Xin Jiang, Zhengdong Lu, Lifeng Shang, Hang Li, Xiaoming Li. Neural Generative Question Answering. Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence (IJCAI-16). 2016.

[24] Wang, D., & Nyberg, E. (2015). A long short-term memory model for answer sentence selection in question answering. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)* (Vol. 2, pp. 707-712).

[25] Hartawan, A., & Suhartono, D. (2015). Using vector space model in question answering system. *Procedia Computer Science*, *59*, 305-311.

[26] Park, S., Kwon, S., Kim, B., & Lee, G. G. (2015). ISOFT at QALD-5: Hybrid Question Answering System over Linked Data and Text Data. In *CLEF (Working Notes)*.

[27] Ben, H., Peter, C., and Hannaneh, H. 2015. Learning Knowledge Graphs for Question Answering.*In Proc. of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies,* Denver, Colorado, pp. 851–861.

[28] Baudiš, P. 2015. YodaQA: a modular question answering system pipeline. *In 19th Int. Student Conf. on Electrical Engineering,*POSTER, pp. 1156-1165.

[29] Nicosia, M., Filice, S., Barrón-Cedeno, A., Saleh, I., Mubarak, H., Gao, W., & Màrquez, L. (2015). QCRI: Answer selection for community question answering-experiments for Arabic and English. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)* (pp. 203-209).

[30] Quan, H. T., Tran, V., Vu, T., Nguyen, M., & Pham, S. B. (2015). JAIST: Combining multiple features for answer selection in community question answering. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)* (pp. 215-219).

[31] Helena, G.A., Grigori, S., Darnes, V., and David, P. 2015. CICBUAPnlp: Graph-Based Approach for Answer Selection in Community Question Answering Task. In *Proc. of the 9th Int. Workshop on Semantic Evaluation (SemEval 2015),* Denver, Colorado, pp. 18–22.

[32] Shizhu, H., Zhang, Y., Liu, K. and Zhao, J.2014. CASIA@V2: A MLN-based QuestionAnswering System over Linked Data. *In CLEF (Working Notes)*, pp. 1249-1259.

[33] Heba, A., Maha, R., Bassant, F., Reham, M., Alaa, M., Nagwa, E.M., and Marwan, R. 2014. Al-Bayan: An Arabic Question Answering System for the Holy Quran. In *Proc. of the EMNLP 2014 Workshop on Arabic Natural Language Processing (ANLP),* Doha, Qatar, pp. 57–64.

[34] Kotaro, S., Hyogo, M., Eisuke, M., Takahisa, J., Hideyuki, S., Tatsunori, M., Madoka, I., Noriko, K. 2014. Forst: Question Answering System Using Basic Element at NTCIR-11 QA-Lab Task. In *Proc. of the 11th NTCIR Conference,* Tokyo, Japan, pp. 532–536.

[35] Junwei, Bao & Duan, Nan & Zhou, Ming & Zhao, Tiejun. (2014). Knowledge-Based Question Answering as Machine Translation. 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014 - Proceedings of the Conference. 1. 967-976. 10.3115/v1/P14-1091.

[36] Di. Wang, L. Boytsov, J. Araki, A. Patel, J. Gee, Z. Liu, E. Nyberg, and T. Mitamura. cmu multiple-choice question answering system at ntcir-11 qa-lab. In Proceedings of the 11th NTCIR Conference, 2014.

[37] Pragisha, K., & Reghuraj, D. P. (2014). A Natural Language Question Answering System in Malayalam Using Domain Dependent Document Collection as Repository. *International Journal of Computational Linguistics and Natural Language Processing*, *3*(3), 2279-0756.

[38] Sean, G., Zadrozny, W., & Avadhani, A. (2014). Watsonsim: Overview of a question answering engine. *arXiv preprint arXiv:1412.0879*.

[39] Abdullah M. Moussa and Rehab F. Abdel-Kader. 2011. QASYO: A Question Answering System for YAGO Ontology. International Journal of Database Theory and Application Vol. 4, No. 2.

[40] Avetisyan, A., and Avetisyan, V. 2010. LOOK4: Enhancement of web search results with Universal Words and WordNet. *In Proceedings of the 5th Int. Conf. on Global WordNet*, Mumbai, India, pp. 1-5.

[41] Cardeñosa, J., Gallardo, C., and de la Villa, M.A. 2009.Interlingual information extraction as a solution for multilingual QA systems. *Flexible Query Answering Systems, 8th International Conference, Lecture Notes in Computer Science,* Vol. 5822, pp. 500-511.

[42] Surve, M., Singh, S., Kagathara, S., Venkatasivaramasastry, K., Dubey, S., Rane, G., Saraswati, J., Badodekar, S., Iyer, A., Almeida, A., Nikam, R., Perez, CG., and Bhattacharyya, P. (2004). Agro-explorer: A meaning based multilingual search engine. In *International Conference on Digital Libraries (ICDL)*.

[43] Radev, D., Fan, W., Qi, H., Wu, H., & Grewal, A. (2002). Probabilistic question answering on the web. In *Proceedings of the 11th international conference on World Wide Web* (pp. 408-419). ACM.

[44] Moldovan, D., Harabagiu, S., Pasca, M., Mihalcea, R., Goodrum, R., Girju, R., & Rus, V. (1999). Lasso: A tool for surfing the answer net. In *TREC* (Vol. 8, pp. 65-73).

[45] Woods, W. A. (1973). Progress in natural language understanding: an application to lunar geology. In *Proceedings of the June 4-8, 1973, national computer conference and exposition* (pp. 441-450). ACM.

[46] Green, F., Bert, Alice, K., Wolf, Chomsky, C., and Laughery, K. 1963. BASEBALL: An automatic question answerer. *In Proceedings of the Western Joint IRE-AIEE-ACM Computing Conference,* Los Angeles, pp. 219-224.

[47] Ask Jeeves. Wikipedia.[Online]. Available: http://en.wikipedia.org/wiki/Ask.com

[48] V. Agarwal, and P. Kumar, "UNLization of Punjabi text for natural language processing applications," *Sādhanā,* 43(6), pp. 1-23, 2018.

[49] V. Agarwal, and P. Kumar, "A Multilingual Cross-Domain Client Application Prototype for UNL-ization and NL-ization for NLP Applications," *Digital Scholarship in the Humanities*, 32(3), pp. 461-475, 2016.

[50] H. Singh, and P. Bhatia, "NLization of adjectives conjunctions determiners and verbs with EUGENE," *M.E dissertation*, Thapar University, Patiala, 2013.

[51] A. Verma and P. Bhatia, "NLization of Nouns, Pronouns, Prepositions and Sentence Structures with EUGENE," *M.E dissertation*, Thapar University, Patiala, 2013.