I.J. Intelligent Systems and Applications, **2015, 12, 57-67**
Published Online November 2015 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijisa.2015.12.06

Keywords based Closed Domain Question Answering System for Indian Penal Code Sections and Indian Amendment Laws

Rohini P. Kamdi
SRCOEM, M.Tech, Computer Science Engineering, Nagpur, India
Email: rohinikamdi29@gmail.com

Avinash J. Agrawal
SRCOEM, Associate Professor, Computer Science Engineering, Nagpur, India
Email: avinashjagrawal@gmail.com

Abstract—In information retrieval, Question Answering (QA) is the task of answering a question posed in natural language (NL) using either a pre-structured database or a collection of natural language documents without human intervention. Question Answering systems are categorized on their available resource for answers. The domain specific Question Answering System gives more exact and correct answers than web based Question Answering system as it is limited for only one domain resource to answer. This paper proposes the closed domain Question Answering System for handling the legal documents of Indian Penal Code (IPC) sections and Indian Amendment Laws to retrieve more precise answers. This system tries to retrieve the exact answers from stored knowledge-base for the query related to Indian Penal Code (IPC) sections and Indian Amendment Laws asked by user. This Keyword based Question Answering System works on structured, unstructured and non-question form queries. The closed domain Question Answering system gives more accurate answer than other open domain system as it restricted single resource. Keywords from both queries and answer corpus play important role for extracting answer.

Index Terms—Question Answering, Information Retrieval Natural language processing, Indian Penal Code (IPC) sections, Indian Amendment laws, keywords and knowledge-base.

I. Introduction

In early 90s, the first Question Answering task in TREC 8 (Text Retrieval Conference) discovered a growing need for more refined search engines able to retrieve the specific information that could be considered as the finest possible answer for the user question. Such systems must go further than document selection, by extracting appropriate part. They should either give the answer if the question is factual or provide a summary if the question is theoretic.

Question Answering (QA) is an area of natural language processing research designed at providing the users with a suitable and natural interface for accessing exact information. The typology of question relied on 13 categories. Each of 13 categories was linked a search strategy of the answer in the knowledge base. The textual database had replaced the knowledge base in the earlier work. An information retrieval based system exploiting only statistic knowledge of the corpus leads to the explanation of a system able to answer less than half of the questions.

The problem intersects two domains: Information Retrieval (IR) and Natural Language Processing (NLP). IR is enhanced by integrating NLP functionalities at a enormous scale, *i.e.* independently of the domain, and necessarily having a huge linguistic coverage. This integration allows the selection of the relevant passages by linguistic features at the syntactic or even semantic level. The possible NL document collections used for QA systems include: a local collection of reference texts, a set of Wikipedia pages and a subset of World Wide Web pages. QA deals with a broad range of question types which includes: fact, list, definition, How, Why, hypothetical, semantically constrained, and cross-lingual questions.

This paper is classified as follows: Section 2 comprises related work in Question Answering, section 3 fallows proposed approach, section 4 gives implementation details, section 5 fallows result and discussion and section 6 gives the conclusion and future work.

A. *Basic Elements of QA System*

QA System has the basic elements as:

1) Question Processing

Question asked by the user is the input to the Question Processing. This captures the semantic of question for what the question is asked by the user. The Question Processing has three tasks as:

a) Determining the question type
b) Determining the answer type

c) Extracting keywords from the question and formulate a query.

a) Question Types

According to the answers, there are five classes of questions as:

Table 1. Classes of Questions

Class 1	Answer: single datum / list of item C: who, when, where, how (old, much, large)
Class 2	A: multi-sentence C: extract from multiple sentence
Class 3	A: across several text C: comparative/contrastive
Class 4	A: an analysis of retrieved information C: synthesized coherently from various retrieved fragment
Class 5	A: result of reasoning C: word or domain knowledge and common sense reasoning

b) Types of QA

According to the domain of answers, there are two types of Question Answering systems as:

Open-Domain QA System: Open-domain question answering deals with the questions about nearly everything, and can depend on common ontologies and world knowledge. Also these systems usually have much more data available for extracting the answer. The system is able to potentially answer any question but has very little precision as the domain is not specific.

Closed-Domain QA System: Closed-domain question answering deals with the questions under a definite domain, and can be seen as an easier task because NLP systems can utilize domain-specific knowledge frequently formalized in ontologies. It has very high precision but requires wide language processing and it is restricted to single domain.

c) Keyword Selection

To give more specific answer, the keywords are helpful for finding the relevant text in question. These keywords can be extended with lexical or semantic alternations for improved matching like, the word “producer” can be taken as “produce”, the phrase “has been sold” can be taken as “sell” for keyword selection. Also various words based on significance like non-stopwords in questions, all complex nominal (plus adjectives), all other nouns, all verbs (not focus on tense), and potential answer type are focused for keywords

2) Document Retrieval

From the selected keywords, a query is formulated and is given to the Passage retrieval component. Here, all the passages are extracted containing the selected keywords. The quality of passage depends upon the loops. It follows some simple heuristic algorithms to decide whether the certain keyword is added or dropped for candidate answering text.

For example, if it uses the initial 6 keywords in the first iteration, it fallows the algorithm like: if the number of passages is less than a threshold then the query is too strict, therefore drop a keyword otherwise if the number of passages is greater than a threshold then query is too relaxed, and therefore add a keyword.

The ranking of passages is done by constructing the keyword windows in which; it searches how many times certain keywords are appeared in the passages. The passage scoring is depends upon

- The number of question keywords obtained in the same sequence in the window.
- The number of keywords separating the most distant keywords in the window.
- The number of unmatched keywords.

More appropriate passage is selected according to passage score for an answer. The passage retrieval module deals with document retrieval from the database for extracting the passage that contains the candidate answer text.

3) Answer Extraction

The representation of the question and the representation of candidate answer bearing texts are matched against each other to give a specific and correct answer in the answer extraction component. From this set of such candidate answers are produced and then ranked according to the likelihood of correctness. The features for answer ranking are:

- Question term numbers matched in the answer passage.
- Question terms numbers matched in the same phrase or sentence as the candidate answer.
- Number of question terms matched, separated from the candidate.
- Number of terms occurring in the same order in the answer passage as in the question.
- Average distance from the candidate answer to the question term matches.

II. Related Work

The open domain QA System [1] described the use of Wikipedia as a rich knowledge source in a question answering system with multiple answer matching modules based on different types of semi-structured knowledge sources of Wikipedia, including article content, infoboxes, article structure, category structure, and definitions. These semi-structured knowledge sources each have their unique strengths in finding answers for specific question types, like as infoboxes for factoid questions, category structure for list questions, and definitions for descriptive questions.

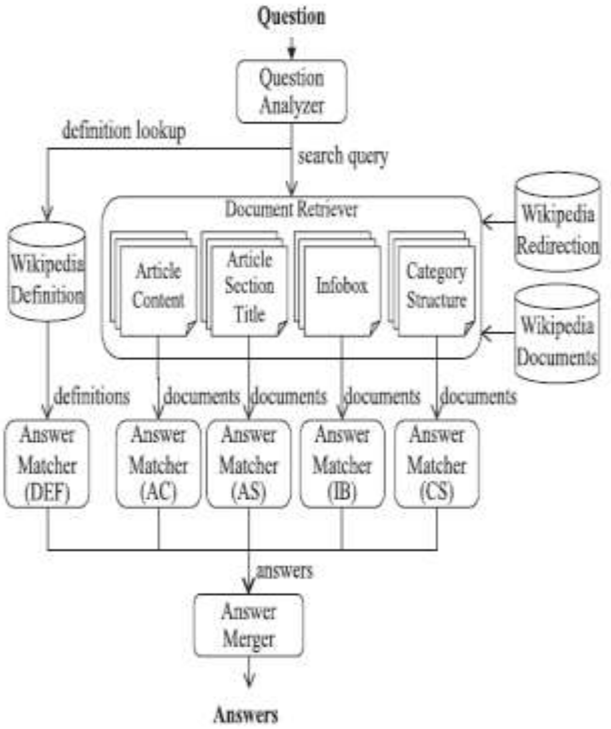


Fig.1. System Overview [1]

In this for Question Analysis, questions in natural language form are analyzed using multiple linguistic analysis techniques, including POS tagging, chunking, and named entity tagging[12] and then analyzed result into the form of answer format (AF), answer theme (AT) and question target (QT). The AF has three possible values as factoid, list, and descriptive, an AT is the class of the object or description sought by the question and A QT consists of two parts as object that the question is about and property of interest that a question attempts to get at regarding the object.

For retrieving an answer, it selects the best answer a for given question q that maximizes the multiplication of question analysis score SQ(r|q), document retrieval score SD(d|r) and the answer matching score SA(M)(a|q,r,d) where r, a, d are question analysis result, answer candidate and retrieved document, respectively and scores are normalized between 0 and 1. The answers extracted from multiple modules are merged using an answer merging strategy that reflects the specialized nature of the answer matching modules. The main motivation behind this work was to devise a way to utilize the existing semi-structured, large-size Wikipedia database as a knowledge source for a QA system without building high-cost knowledge base.[1]

For semi-structured knowledge-bases QA System [2], a new architecture to develop a factoid question answering system based on the DBpedia ontology and the DBpedia extraction framework. Dbpedia is a project that aims at extracting information based on the semi-structured data presented within the Wikipedia articles. This system is divided into three parts as Question Classification and Decision Model Generation**,** Question Processing and Query Formulation and Execution.

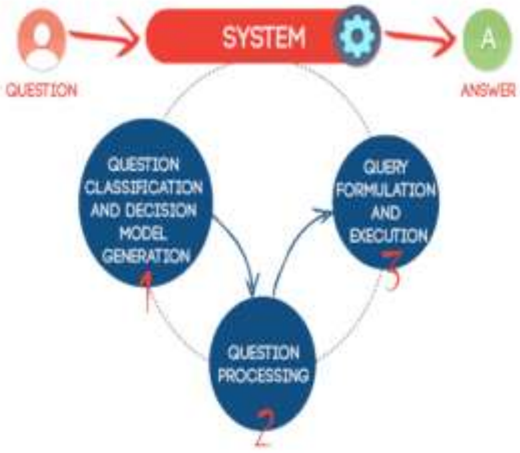

QUESTION
SYSTEM
A
ANSWER
QUESTION CLASSIFICATION AND DECISION MODEL GENERATION
1
QUESTION PROCESSING
2
QUERY FORMULATION AND EXECUTION
3

Fig.2. Global Question Answering System Architecture [2]

This paper [2] is divided into 3 parts as Question Classification and Decision Model Generation**,** Question Processing and Query Formulation and Execution.

(i) Question Classification and Decision Model Generation: It is given by two components; first component is the question classifier which pre-processes the question dataset and trains it by the SVM algorithm [2], [11] which is a binary classifier giving two classes as the coarse-grained classes and the fine-grained classes for their proposed QA System. And the second component is the decision model generator which fallows tokenization, stop words removal and features extraction using the bag of words. Then it creates trained and tested file for estimating accuracy.

(ii) Question Processing: It allows identifying the question type of given question by extracting resources using DBpedia spotlight tool and extracting keywords using processing.

(iii) Query Formulation: It involves Ontology Class Determination, which determines ontology classes and their properties used to construct the final SPARQL query and the result of the query is an RDF file holding ontology classes and properties, Query formulation used to retrieve the answer from DBpedia composed from resources and ontology classes determined by the keyword set and Execution in which the system interrogates the DBpedia Server to get a response as an RDF file and then parsed to get the answer of the given question.

Authors in [3] given the system which finds answers of Malayalam factual questions by analyzing a repository of Malayalam documents for handling the four classes of factual questions in Malayalam for closed domain. The QA system is divided into three modules as Question Analysis, Text Retrieval and answer snippet extraction and Answer identification.

(i) Question Analysis: It takes single sentence level questions as an input. The aim of this module is to identify the question word, the query and expected set of answer templates and fallows the NLP algorithm for preprocessing.

(ii) Text Retrieval and answer snippet extraction: Based on the query words the answer candidates are retrieved from the document collection for answer identification.

The document collection is indexed and, which has total keyword match with the question are selected for answer snippet extraction. For this, it checks the count of match of the query with each sentence. The sentences which have a fuzzy match to the query are selected as the answer candidates are represented using a triplet containing the sentence, index and count of the match. The index is used to extract the actual sentence. The index value is assigned at the time of text splitting and count of the match gives the value of match with the question. These answer candidates are passed to the next module selection of answer candidates,

(iii) Answer identification: It has two sub-modules as, Scoring and Ranking of the answer candidates which performs the scoring and ranking, and selects the winner candidate using matching window sizes. This answer candidate which has the highest score is selected as the winner candidate and this snippet is further processed for answer extraction. And the second is Answer Extraction using Named Entity Recognition in which the expected named entity of the question is identified by analyzing the question word and then nearest surrounding words of the question word are analyzed to identify the expected answer entity.

Jibin Fu in the paper [4] proposed a music knowledge question answering system on the ontology knowledge base through which the users can ask a question about music knowledge in natural language, and the system automatically extracts relative knowledge to give answer based on FAQ and ontology knowledge base. It has three processes as Question Classification, FAQ and Question Analyzer and Answer Extraction.

(i) Question Classification: It uses the ontology and improved Bayesian-based method [15]. First, the concepts in user's questions are extracted in support of ontology knowledge base and then the frequency of terms calculated using "word-bag" model for finding class of question.

(ii) FAQ and Question Analyzer: Frequently asked questions are stored in FAQ module which can quicken the processing. The similarity of user's question and question in FAQ candidate question set is computed. If user's question can't match in the FAQ, the question is transferred to question analyzer module in which, question template method is used to extract semantic representation for a simple question and for complex question and abnormity question; keyword association method is used for probability of semantic representation. For each template, its semantic representation is extracted, once a question can match a question template, the semantic representation of the question can be located.

(iii)Answer Extraction: It fallows two strategies: In first, one directly match question with the question in FAQ, for frequently asked question, and in second strategy, for a question not included in FAQ, analyze the question and extracts the answer in support of ontology and logic reasoning. First has higher priority than strategy two.

The relative concepts are extracted from ontology and the relations between concepts are reason and knowledge point is extracted to form answer.

III. Proposed Work

Using the literature survey of Question Answering Systems, we can say can that the closed domain QA System is more accurate than the open domain QA System as it only works on single domain to give specific answer for user query.. If we see scenario of queries related to legal documents of IPC sections and different Indian laws, there is no such QA system, which ensures the correct answers. The user generally asks the query in the non question form, for example "If Ram killed Shyam, then punishment to Ram". Also the same query in unstructured form as "Charges for murder". And for the structured query as "What is the punishment for murder?", all the three questions are asked in different forms but give the answer related to punishment for murder. So, the idea to develop the closed domain Question Answering System for IPC Sections and Indian Laws is proposed to give more specific and relevant answer for user's query on this domain.

The system design flow can be depicted as:

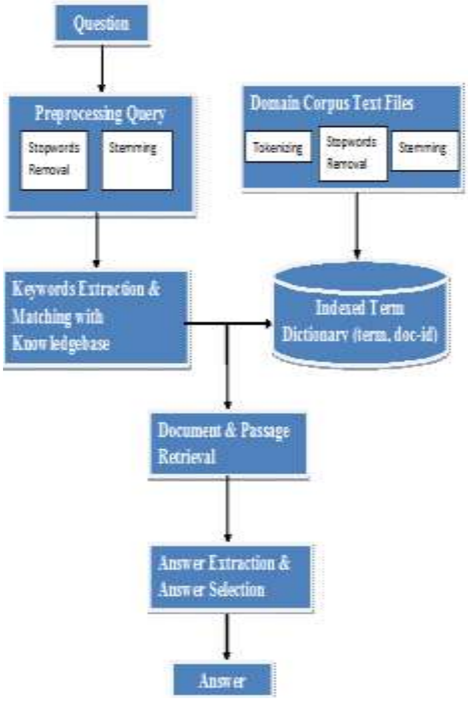

Question
Preprocessing Query
Stopwords Removal
Stemming
Domain Corpus Text Files
Tokenizing
Stopwords Removal
Stemming
Keywords Extraction & Matching with Knowledgebase
Indexed Term Dictionary (term, doc-id)
Document & Passage Retrieval
Answer Extraction & Answer Selection
Answer

Fig.3. Proposed System Flow

IV. Implementation Details

A) *Corpus*

The most important task for designing the closed domain Question Answering System is to decide the domain that is resource on which the questions asked for specific answers. There are so many QA systems already present for different closed domains. So we are dealing with the new domain for answering the user queries on IPC sections and some Indian Laws. The legal documents of IPC sections and laws like parent and company amendment are necessary to know in different ways for different users. On different websites the IPC sections and laws documents are available for interested users. But for exact answers on different questions related these documents can be given by using QA system. So we have gone through the different websites and taken the text data from the websites for developing our corpus:

As these documents are authorized and highly sensitive therefore the legality is referenced by the lawyer. These websites are only used get the resource data. We have taken the data in text format to create the corpus. For each IPC sections and different laws, there is one text file, such for 511 IPC sections, 511 text files are stored and also for constitution amendment law 94 text files and each for parent amendment law and company amendment law are stored. This corpus is used as resource i.e. knowledgebase for final answer extraction.

B) *Preprocessing*

For information retrieval, the preprocessing of text data is necessary to deal with the exact required data. Preprocessing of text data can be done using various tools. Here, for our corpus we have used the Rapidminer tool to preprocess the text files.

Rapidminer is the tool is used for text mining with different features of machine learning processes. For preprocessing our text data corpus, the certain flow is followed by Rapidminer as:

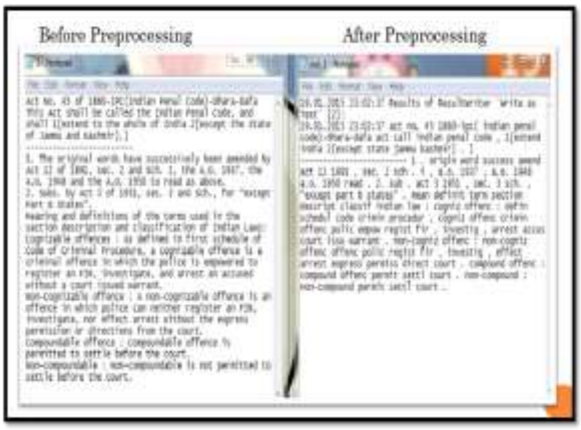
- Text document corpus
- Data preprocessing
  - Read text document
  - Select Process Document Operator
  - Tokenization
  - Filtering Stop words
  - Stemming
  - Writing result as text

Major tasks in preprocessing are tokenization, stopwords removal and stemming.

Tokenization: Perform linguistic tokenization which includes each word and also the numbers as single token.

Stopwords Removal: The English stopwords like "is, for, the, in, etc." are removed from each text files of corpus using Rapidminer.

Stemming: Stemming is most important process by which the different forms of word is replaced with basic root word. Rapidminer uses Porter Stem algorithm to perform stemming.



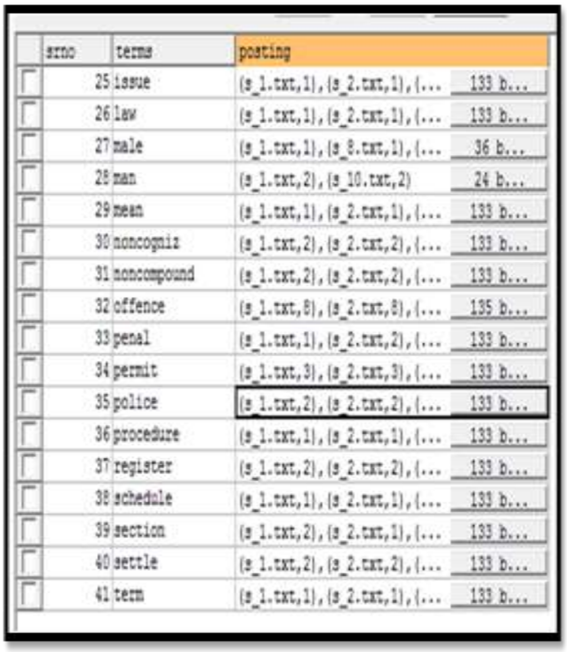
```
Act No. 45 of 1860-IPC[Indian Penal Code]-dhara-dafa
This Act shall be called the Indian Penal Code, and
shall 1[extend to the whole of India 2[except the State
of Jammu and Kashmir].]
------------------------
1. The original words have successively been amended by
Act 12 of 1891, sec. 2 and Sch. I, the A.O. 1937, the
A.O. 1948 and the A.O. 1950 to read as above.
2. Subs. by Act 3 of 1951, sec. 3 and Sch., for "except
Part B States".
Meaning and definitions of the terms used in the
section description and classification of Indian Laws:
Cognizable offences : as defined in first schedule of
Code of Criminal Procedure, a cognizable offence is a
criminal offence in which the police is empowered to
register an FIR, investigate, and arrest an accused
without a court issued warrant.
Non-cognizable offence : a non-cognizable offence is an
offence in which police can neither register an FIR,
investigate, nor effect arrest without the express
permission or directions from the court.
Compoundable offence : compoundable offence is
permitted to settle before the court.
Non-compoundable : non-compoundable is not permitted to
settle before the court.
```

```
29.05.2015 23:02:37 Results of ResultWriter 'Write as
Text' [2]:
29.05.2015 23:02:37 act no. 45 1860-ipc[ indian penal
code]-dhara-dafa act call indian penal code , 1[extend
india 2[except state jammu kashmir] . ]
------------------------ 1 . origin word success amend
act 12 1891 , sec. 2 sch . 1 , a.o. 1937 , a.o. 1948
a.o. 1950 read . 2. sub . act 3 1951 , sec. 3 sch . ,
"except part b states" . mean definit term section
descript classif indian law : cognit offenc : defin
schedul code crimin procedur , cognit offenc crimin
offenc polic empow regist fir , investig , arrest accus
court issu warrant . non-cognit offenc : non-cognit
offenc offenc polic regist fir , investig , effect
arrest express permiss direct court . compound offenc :
compound offenc permit settl court . non-compound :
non-compound permit settl court .
```

Fig.4. Input and Output Text files of Preprocessing

C) *Indexed Term Dictionary*

The preprocessed output text file contains only stemmed words which can be considered as the key terms for the closed domain of IPC sections and amendment laws. As the corpus has more than 600 text files, it is necessary to know which key terms are present in which document for document retrieval. So, the Indexed Term Dictionary is created by using java and stored as table in Mysql. The dictionary forms structure like inverted index containing two columns as "term" and "posting". Posting represents two parts as "document id" and "frequency count of term in document". The dictionary contains nearly 7000 different words with its count in specific document. This Indexed Term Dictionary is used for retrieving the document ids containing the specific keywords in information retrieval for answer extraction.



	srno	terms	posting	
□	25	issue	(s_1.txt,1),(s_2.txt,1),(...	133 b...
□	26	law	(s_1.txt,1),(s_2.txt,1),(...	133 b...
□	27	male	(s_1.txt,1),(s_8.txt,1),(...	36 b...
□	28	man	(s_1.txt,2),(s_10.txt,2)	24 b...
□	29	mean	(s_1.txt,1),(s_2.txt,1),(...	133 b...
□	30	noncogniz	(s_1.txt,2),(s_2.txt,2),(...	133 b...
□	31	noncompound	(s_1.txt,2),(s_2.txt,2),(...	133 b...
□	32	offence	(s_1.txt,8),(s_2.txt,8),(...	135 b...
□	33	penal	(s_1.txt,1),(s_2.txt,2),(...	133 b...
□	34	permit	(s_1.txt,3),(s_2.txt,3),(...	133 b...
□	35	police	(s_1.txt,2),(s_2.txt,2),(...	133 b...
□	36	procedure	(s_1.txt,1),(s_2.txt,1),(...	133 b...
□	37	register	(s_1.txt,2),(s_2.txt,2),(...	133 b...
□	38	schedule	(s_1.txt,1),(s_2.txt,1),(...	133 b...
□	39	section	(s_1.txt,2),(s_2.txt,1),(...	133 b...
□	40	settle	(s_1.txt,2),(s_2.txt,2),(...	133 b...
□	41	term	(s_1.txt,1),(s_2.txt,1),(...	133 b...

Fig.5. Indexed Term Dictionary (Terms, Doc_id, freq)

D) Question Dataset

As we are dealing with the closed domain Question Answering, the system need to answer the different questions related to domain. So the dataset of 200 different questions on IPC sections and amendment laws is maintained to train our QA system. And also 100 questions for testing the system. These questions are taken from different users by survey, so the system can handle structured as well as unstructured queries from user. According to different trained questions, the QA system is designed to give the answers for those questions correctly.

E) Information Retrieval

The design "model'' of QA is given from user query to passage and document retrieval. It has 3 components as:

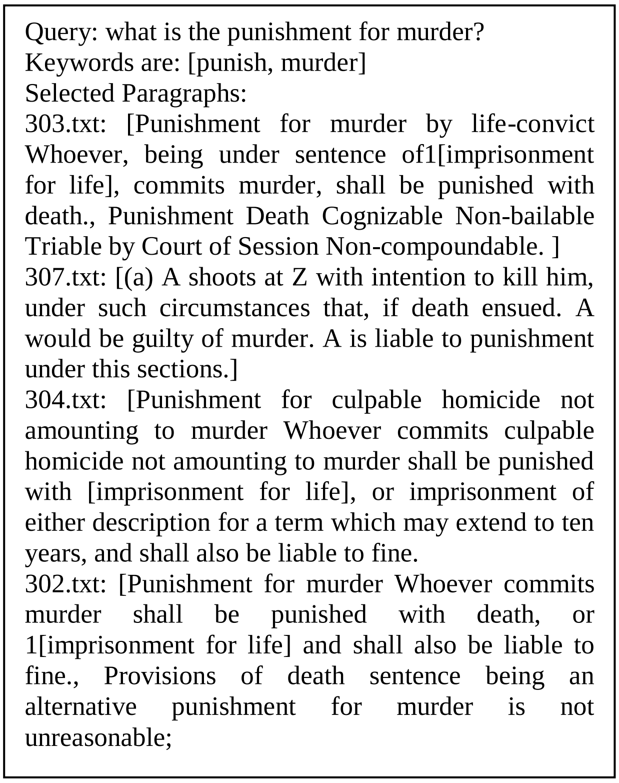
(i)User query: User will enter the query related to IPC Sections and amendment laws. This query can be structured, non question or in unstructured form. For example, the user can ask the question "What is the punishment for murder?" or "If Ram killed Shyam, then charges on Ram" or "which IPC sections applied on offence murder?" or "killing charges " etc.

(ii)Extract Keywords: From the user query, the keywords are extracted. These keywords are obtained by removing the symbols and stopwords from user query. Also the stemming is applied on keywords so as to match with term indexed dictionary terms for document retrieval. English stopwords and stemmed words dictionary is maintained for extracting the keywords. These keywords have the most important role in our Question Answering System as these are used for answer extraction process.

(iii)Document and Passage Retrieval: The keywords so obtained from query are matched in Term Indexed Dictionary to find the document ids in which these keywords are present. For more than one keywords, it take the intersections of all document ids where these terms are present so that where all the keywords are present only those document are to be retrieved for candidate answer passages.

Now, in the extracted documents, the positions of each terms is maintained so that to find the position of keywords. The positive and negative threshold difference value can be given to extract the paragraph from documents. The keywords that obtained from user query are matched in documents and where it meets with threshold value, it gives the certain passage from our text file corpus. It can give one or more paragraphs which satisfies the keywords matching. Here, we are using the proxitimity value as 3. The passages or lines are selected form certain documents are only those which are retaining the proxitimity of value 3 with extracted keywords.

The example for certain queries with its document and passage retrieval is given as:



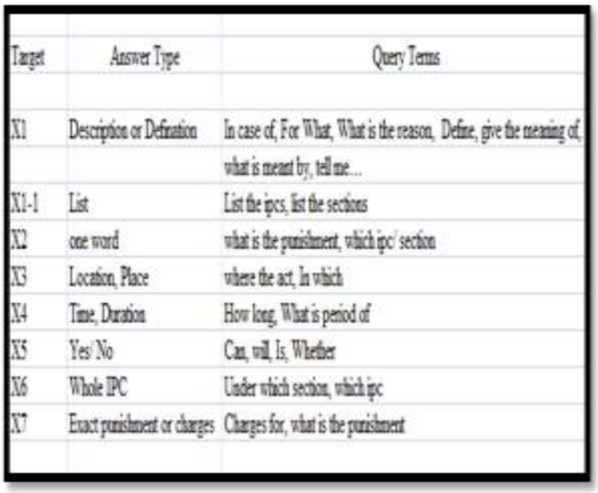
Query: what is the punishment for murder?
Keywords are: [punish, murder]
Selected Paragraphs:
303.txt: [Punishment for murder by life-convict Whoever, being under sentence of1[imprisonment for life], commits murder, shall be punished with death., Punishment Death Cognizable Non-bailable Triable by Court of Session Non-compoundable.]
307.txt: [(a) A shoots at Z with intention to kill him, under such circumstances that, if death ensued. A would be guilty of murder. A is liable to punishment under this sections.]
304.txt: [Punishment for culpable homicide not amounting to murder Whoever commits culpable homicide not amounting to murder shall be punished with [imprisonment for life], or imprisonment of either description for a term which may extend to ten years, and shall also be liable to fine.
302.txt: [Punishment for murder Whoever commits murder shall be punished with death, or 1[imprisonment for life] and shall also be liable to fine., Provisions of death sentence being an alternative punishment for murder is not unreasonable;

Fig.6. Information Retrieval

F) Answer Extraction and Exact Answer Selection

After retrieving the suitable passages for candidate answers for the user query, Answer Extraction is the most important component of Question Answering System. How to obtain the exact answer that is selection of the answer is major task for dealing with the QA system. It needed some extra knowledge about both questions and expected answers for retrieving and selecting the exact answer. As we studied, there are many types of questions handled by the Question Answering System, but here we are using the Question Classification according to the answer type. The answer to the queries can be description, definition, list, yes or no, exact one specific word or line or the whole document. The answer to the user query for our domain that is the IPC sections and amendment laws can description about any section, definition of any related term, list of applied IPCs for related crime, yes or no about any section or related query, exact punishment for certain crime or whole IPC section.

For these different answer types, the question can be classified accordingly to give certain "target value" for question classification as:



Target	Answer Type	Query Terms
X1	Description or Defination	In case of, For What, What is the reason, Define, give the meaning of, what is meant by, tell me...
X1-1	List	List the ipcs, list the sections
X2	one word	what is the punishment, which ipc/ section
X3	Location, Place	where the act, In which
X4	Time, Duration	How long, What is period of
X5	Yes/ No	Can, will, Is, Whether
X6	Whole IPC	Under which section, which ipc
X7	Exact punishment or charges	Charges for, what is the punishment

Fig.7. Question Classification according to Answer Type

Here we consider the “Query terms” to determine the “Target value” for classifying the questions according to the expected answer for certain query.

For example, the query terms like “what is the reason”, “In case of”, “suggest or tell me” gives answer as description. The terms like “define” or “what is meant by” refers the definition as answer. Also the terms like “list the IPCs” or “list the sections” give the expected answer as the list of related IPC sections for related crime. The questions started with the terms like “can. Is, whether” should give the answer as yes or no. Also if user wants the whole IPC section for reference the terms so considered as “which IPC” or “under which section or the term IPC with digit value like “IPC 29”. Also if user wants to know about exact punishment or charges for certain crimes it is referred with the query terms like “what is the punishment” or “charges for”. As we are dealing with the domain of IPC sections, user need to deal with the different punishments and related IPCs mostly. Therefore this exact punishment question has more weight while designing and training the system.

We have different query terms according to the expected answers which are stored with certain target value in MySql table format. For example “X1” is the target value for expected answer as “description” or “definition”, so the different possible query terms for this as show in figure 7 are stored for the same value as “X1” and same for the other answer types. Now, the user query that we have already used in Information Retrieval Modal is going to get used again to deal with classifying the question and determining the target value.

Jaccard Coefficient mechanism is used for nearest matching of the two strings. It gives the relative score of matching the terms in string 1 with the string 2. Jaccard coefficient uses the formula to obtain the score given as:

$$\text{Score} = \text{Query} \cap \text{Query type} / \text{Query U Query type}$$

Here, “Query” from user is considered as String 1 and “Query type” that we have stored with its target value in table is considered as String 2. Now the query from user is going to match with all the query terms from the table and give the score by using the Jaccard formula above. So we have different scores for each query terms depend upon the matching keywords in both strings. To determine the target value for question classification here we considered the maximum score between the user query and query terms so that the target value against the query term is assigned to the user query to give the answer.

The target value determination using Jaccard Coefficient is given as:

```
query: what is the punishment for murder?
keywords
jaccard function :
query :what is the punishment for murder      qtype :In case of
intersec :0  union :9   score :0.0
query :what is the punishment for murder      qtype :What is the
reason
intersec :3  union :7   score :0.42857143
query :what is the punishment for murder      qtype :how to define
intersec :0  union :9   score :0.0
query :what is the punishment for murder      qtype :what defines
intersec :1  union :7   score :0.14285715
query :what is the punishment for murder      qtype :define the word
intersec :1  union :8   score :0.125
query :what is the punishment for murder      qtype :In which
section
intersec :0  union :9   score :0.0
query :what is the punishment for murder      qtype :What is the
intersec :3  union :6   score :0.5
query :what is the punishment for murder      qtype :list the ipc
intersec :1  union :8   score :0.125
.
.
.
.
query :what is the punishment for murder      qtype :What if
intersec :1  union :7   score :0.14285715
query :what is the punishment for murder      qtype :which offence
intersec :0  union :8   score :0.0
query :what is the punishment for murder      qtype :suggest me the
intersec :1  union :8   score :0.125
query :what is the punishment for murder      qtype :Is there any
intersec :1  union :8   score :0.125
query :what is the punishment for murder      qtype :charges for
intersec :1  union :7   score :0.14285715
query :what is the punishment for murder      qtype :what is the
punishment
intersec :4  union :6   score :0.6666667

Maximum Jaccard score=0.6666667
Question Type =        what is the punishment
Target is X7
```

Fig.8. Determination of target Value

Now we have documents with related passages for candidate answer and the target value for the certain question by user. Answer extraction is carried out by using these two components. According to answer type the target value is classified for question and the answer is selected from retrieved passages form documents.

For different type of expected answer to the user query about our domain, we have defined some rules for selecting the answer. For definition or the description about any section or punishment, we are extracting the passage from first file retrieved in the Information Retrieval modal as the corpus we have stored of text files are in sorted ascending order so that most appropriate and matching answer and exact definition will obtained in first passage only. While listing the IPCs or sections for any crime, we only require the related IPCs number value. So to give the related IPC list, only document ids are used as the text files for related IPCs are stored with their section number. If the user will ask any question for which answer can be positive or no then the system will try to return the positive answer as related line from extracted passages if matches with the certain domain constraints otherwise give no answer. If the user wants the whole IPC as the answer then the whole text file related IPC section is retrieved as the text files in corpus are stored with their names and each file has the title with its related contents for example “DharaDafa 300-Murder”.

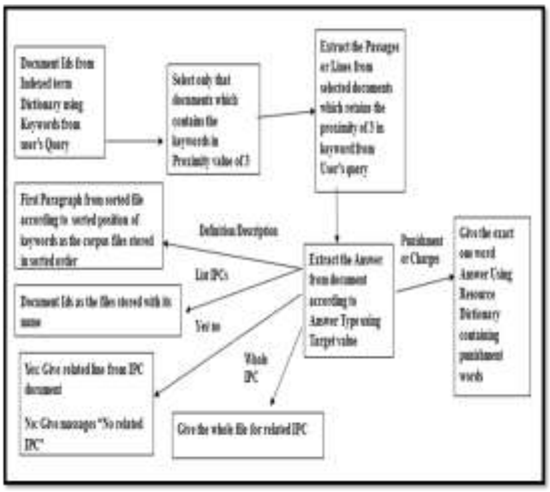

Document Ids from Indexed term Dictionary using Keywords from user's Query
Select only that documents which contains the keywords in Proximity value of 3
Extract the Passage or Lines from selected documents which retains the proximity of 3 in keyword from User's query
First Paragraph from sorted file according to sorted position of keywords as the corpus files stored in sorted order
Definition Description
Extract the Answer from document according to Answer Type using Target value
Punishment or Charges
Give the exact one word Answer Using Resource Dictionary containing punishment words
List IPCs
Document Ids as the files stored with its name
Yes/ no
Yes: Give related line from IPC document
No: Give messages "No related IPC"
Whole IPC
Give the whole file for related IPC

Fig.9. Answer Selection According to Answer Type

Now, as earlier told we are majorly dealing with the questions like “What is the punishment for” or “charges for”. This question needs to give the answer as exact punishment or charges for certain crime. To deal with this type of question, additional knowledge about the answer is required. For that we are maintaining the domain resource dictionary. The Domain Resources Dictionary is the file in which all the possible punishments mentioned in all the sections are stored. For example: death, imprisonment for, liable to fine, solitary confinement etc. So while giving the answer to query about exact punishment or charges about any crime, from the passages or line that we have retrieved in the information retrieval model are searched to obtain such punishment terms stored in dictionary. And give the exact answer to the user query about punishments.

G) Domain Concepts Dictionary

As we are dealing with the keyword based retrieval, therefore whole system depends upon the keywords obtained from user query and the keywords we stored in Indexed Term Dictionary from the corpus text files. The user can give the unstructured query or non question form query to the system. Also the common users are mostly unaware about the different terms which are exactly related to our domain that is IPC section and amendment laws. So we are maintaining the domain concepts dictionary in which the most commonly used words for queries are stored against the related keyword of our domain so that while dealing with the exact keywords it can be referred to the domain specific term.

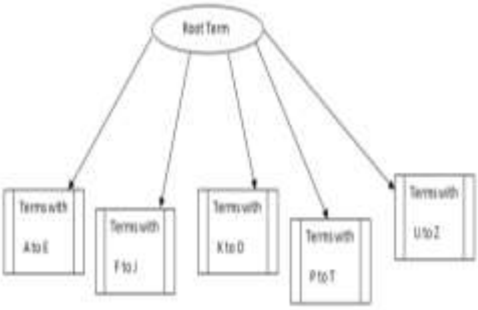


Fig.10. Domain Concepts Dictionary

When user give any word which is not related to domain but is the synonym for the keyword in Indexed Term Dictionary, then this Domain Concepts Dictionary will help to get the root word which the related term of our domain. Such different terms which are commonly used are stored with their different synonyms in structured format. In this, for the single root term different ontology words are given in the alphabetical order. We have divided the dictionary in five parts as: terms with A to E, F to J, K to O, P to T and U to Z. So the searching time for the root term gets reduced as the traversing of synonyms is fallowed in alphabetical order by reading the first character from the keyword.

V. Result And Discussion

We have trained the system with 200 different questions. These questions include various queries by the users about the all 511 IPC sections and Amendment laws. These questions are mostly of structured question format. Out of 200 questions, nearly 100 are of exact punishment type, 45 are of description and definition type, 35 of list answers and nearly 20 are of yes or no type. For these 200 questions, system produces correct output with great accuracy as the queries are structured.. Now, we have again tested the system with different 100 random questions related to our domain. These questions are majorly unstructured and non question form. For these questions the expected answers can be same as of trained questions. Here, we are getting less accuracy than in trained questions. Mostly the questions for yes or no answer have no response during testing because while dealing with keywords the “no” answer cannot be retrieved.

Out of 100 tested questions, we got 87 responses from the system and out of that 87 responses 82 responses for the different queries are correct. We are measuring the correctness of the system by precision and recall. The precision of system is how many correct answers out of total responses by system and the recall is how many correct answers out of total questions tested on system. And the F-score is Harmonic Mean of precision and recall.

So the precision, recall and F-score are calculated using the formula:

$$\text{Precision (P)} = \text{correct answers} / \text{total responses} * 100$$

$$\text{Recall (R)} = \text{correct answers} / \text{total questions} * 100$$

$$\text{F-score} = (2*P*R/ P+R) * 100$$

For our QA system, the precision is 94%, recall is 82% and F-score is 87.59%

Precision	Recall	F-Score
94%	82%	87.59%

Query: what is forgery?
Preprocessed Query [forgery]
Target value using Jaccard function:
Query value =what is
Target is x1
Keywords are: [forge]
Candidate Documents Are:
[463.txt, 469.txt, 468.txt]
Selected Paragraphs:
463. txt_ [Forgery Whoever makes any false documents or false electronic record or part of a document or electronic record, with intent to cause damage or injury, to the public or to any person, or to support any claim or title, or to cause any person to part with property, or to enter into any express or implied contract, or with intent to commit fraud or that fraud may be committed, commits forgery.
468.txt_ [Forgery for purpose of cheating Whoever commits forgery, intending that the 1[document or Electronic Record forged] shall be used for the purpose of cheating, shall be punished with imprisonment of either description for a term which may extend to seven years, and shall also be liable to fine.
469. txt_ [Forgery for purpose of harming reputation Whoever commits forgery, 1[intending that the document or Electronic Record forged] shall harm the reputation of any party, or knowing that it is likely to used for that purpose, shall be punished with imprisonment of either description for a term which may extend to three years, and shall also be liable to fine.
Answer: [Forgery Whoever makes any false documents or false electronic record or part of a document or electronic record, with intent to cause damage or injury, to the public or to any person, or to support any claim or title, or to cause any person to part with property, or to enter into any express or implied contract, or with intent to commit fraud or that fraud may be committed, commits forgery.

Fig.11. Response to Definition Type Question

Here we can study some examples of different questions and their responses from our system for more understanding and to know the working and output of the Question Answering System for IPC Sections and Indian Amendment Laws.

Query: list the ipc for punishment for murder
Preprocessed Query: [ipc, punishment, murder]
Target value using Jaccard function:
Query value =list the ipc for punishment
Target is x1-2
Keywords are: [punish, murder]
Candidate Documents Are:
[303.txt, 113.txt, 109.txt, 302.txt, 111.txt, 108.txt]

====Related IPC====
IPC 108
IPC 109
IPC 302
IPC 303

Fig.12. Response to List Type Question

Query: what is the punishment for theft?
Preprocessed Query [punishment, theft]
Target value using Jaccard function:
Query value =what is the punishment
Target is x11
Keywords are: [punish, theft]
Candidate Documents Are:
[381.txt, 382.txt, 380.txt, 379.txt]
Selected Paragraphs:
379. txt_ [Punishment for theft Whoever commits theft shall be punished with imprisonment of either description for a term which may extend to three years, or with fine, or with both.
380.txt_ [punishment for Theft in dwelling house Whoever commits theft in any building, tent or vessel, which building, tent or vessel is used as a human dwelling, or used for the custody of property, shall be punished with imprisonment of either description for a term which may extend to seven years, and shall also be liable to fine.
381. txt_ [punishment for Theft by clerk of property in possession of master Whoever, being a clerk or servant, or being employed in the capacity of a clerk or servant, commits theft in respect of any property in the possession of his master or employer, shall be punished with imprisonment of either description for a term which may extend to seven years, and shall also be liable to fine.
382.txt_ [punishment for Theft after preparation made for causing death, hurt or restraint in order to the committing of the theft Whoever commits theft, having made preparation for causing death, or hurt, or restrain, or fear of death, or of hurt, or of restraint, to any person, in order to the committing of such theft, or in order to the effecting of his escape after the committing of such theft, or in order to the retaining of property taken by such theft, shall be punished with rigorous imprisonment for a term which may extend to ten years, and shall also be liable to fine.
Answer: [Punishment for theft Whoever commits theft shall be punished with imprisonment of either description for a term which may extend to three years, or with fine, or with both.

Fig.13. Response to Exact Punishment Type Question

As like different 200 questions are trained and different 100 questions are tested on the system.

VI. Conclusion And Future Work

Question Answering requires more complex NLP techniques compared to other forms of Information Retrieval. QA Systems can be developed for resources like web, semi-structured and structured knowledge-base. The Closed Domain QA Systems give more accuracy in finding answers but restricted to single domain only and also require some extra information to deal with the exact answers about the domain.

The QA system for closed domain of legal documents of IPC sections and Indian Laws using machine learning approach and information retrieval is proposed to give the accurate and suitably more correct answers for user's structured, unstructured and non question form queries efficiently. Structured queries have more accuracy than unstructured and non question form. The system face some complexity for dealing with Yes/ No type questions as the answer for such type questions need to be written in proper format telling the exact answer. The precision and recall of the system are 94% and 82% respectively.

To deal with keywords based QA system, we require additional knowledge about the resource domain and also need to maintain the domain terms synonyms dictionary to retrieve the best suitable correct answer for user queries.

For our system, we are only dealing with the keywords both from the user query and the domain corpus. So the semantic nature of question or sentence is not considered. For example, the query "If A gives bribe to B, then charges for A" gives the same response as for the query "If A gives bribe to B then charges for B". Though the query is different, as punishment for the both person will be different but it produces the answer about bribe for both queries because we have not concerned with the semantic of sentence. To add the semantics of questions or sentences, more machine learning applications will be needed. In future, this semantic approach for questions can be applied over the system.

Fast information retrieval is the major application of this system because we are retrieving the answers according to different answer types. Also this system can be applied to other domains like medical or tourism for question answering purpose by replacing the domain corpus.

ACKNOWLEDGMENT

The authors are grateful to the Head, Computer Science and Engineering Department and the Principal, Shri Ramdeobaba College of Engineering and Management, Nagpur for providing adequate facilities to conduct a research. Authors are thankful to the faculty members of Computer Science and Engineering Department for their support and cooperation during this work.

REFERENCES

[1] Pum-Mo Ryu, Myung-Gil Jang and Hyun-Ki Kim. 2014. "Open domain question answering using Wikipedia-based knowledge model." In Information Processing and Management 50 (2014) 683–692, Elsevier.

[2] Adel Tahri and Okba Tibermacine. "DBPEDIA BASED FACTOID QUESTION ANSWERING SYSTEM." In International Journal of Web & Semantic Technology (IJWesT) Vol.4, No.3, July 2013.

[3] Pragisha K. and Dr. P. C. Reghuraj, "A Natural Language Question Answering System in Malayalam Using Domain Dependent Document Collection as Repository." International Journal of Computational Linguistics and Natural Language Processing Vol 3 Issue 3 March 2014 ISSN 2279 – 0756.

[4] Jibin Fu, Keliang Jia and Jinzhong Xu, "Domain Ontology Based Automatic Question Answering", 2009 International Conference on Computer Engineering and Technology.

[5] Anette Frank , Hans-Ulrich Krieger, Feiyu Xu, Hans Uszkoreit, Berthold Crysmann, Brigitte Jörg and Ulrich Schäfer, "Question answering from structured knowledge sources", In German Research Center for Artificial Intelligence, DFKI, Stuhlsatzenhausweg 3, 66123 Saarbrücken, Germany Available online 27 January 2006.

[6] Perera, Rivindu (2012) "IPedagogy: Question Answering System Based on Web Information Clustering", In Proceedings of the 2012 IEEE Fourth International Conference on Technology for Education (T4E '12). IEEE Computer Society, Washington, DC, USA.

[7] Menaka S and Radha N. "Text Classification using Keyword Extraction Technique", in International Journal of Advanced Research in Computer Science and Software Engineering, Volume 3, Issue 12, December 2013.

[8] MatthewW. Bilotti and Eric Nyberg," Improving Text Retrieval Precision and Answer Accuracy in Question Answering Systems", the 2nd workshop on Information Retrieval for Question Answering (IR4QA), pages 1–8 Manchester, UK. August 2008.

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

[10] Eric Brill, Susan Dumais and Michele Banko, "An Analysis of the AskMSR Question-Answering System", Conference on Empirical Methods in Natural Language Processing (EMNLP), Philadelphia, July 2002, pp. 257-264. Association for Computational Linguistics.

[11] Yaoyong Li, Kalina Bontcheva, and Hamish Cunningham, "SVM Based Learning System for Information Extraction", Department of Computer Science, the University of She_eld, She_eld, S1 4DP, UK.

[12] Moussa, Abdullah M. & Rehab, Abdel-Kader (2011) "QASYO: A Question Answering System for YAGO Ontology". International Journal of Database Theory and Application. Vol. 4, No. 2, June, 2011. 99.

[13] W.A.Woods, R.M. Kaplan, B.L. Nash-Webber, "The Lunar sciences natural language information system: Final report", Technical Report BBN Report 2378, Bolt Beranek and Newman Inc., Cambridge, MA, 1972.

[14] Lee, C., Hwang, Y.-G., Oh, H.J., Lim, S., Heo, J., Lee, C.-H., et al (2006). "Fine-grained named entity recognition using conditional random fields for question answering". In Proceedings of Asia information retrieval symposium (pp. 581–587).

[15] ZHANG Yu, LIU Ting, WEN Xu, "Modified Bayesian Model Based Question Classificatio", 2005, vol.19, pp. 100-105.

[16] Amit Mishra, Nidhi Mishra and Anupam Agrawal, "Context-Aware Restricted Geographical Domain Question Answering System", In 2010 International Conference on Computational Intelligence and Communication Networks.

[17] Rohini P. Kamdi and Dr, A. J .Agrawal, "Domain Specific Question Answering System", in International Journal of Electrical, Electronics And Computer Systems (IJEECS), Vol 4, Issue-2, 2015.

[18] Rohini P. Kamdi and Dr, A. J .Agrawal, "Closed Domain Question Answering System for IPC Sections and

Amendment Laws" in The International Journal For Engineering Applications and Technology (IJFEAT), Volume-2, Issue-1, 2015.

Authors' Profiles

Rohini P. Kamdi has received her B.E. degree in Computer Science and Engineering from Gowindrao Wanjari College of Engineering and Technology, Nagpur in 2012. She is pursuing Masters in Technology in Computer Science from Shri Ramdeobaba College of Engineering and Management, Nagpur. Her research interest includes information retrieval and machine learning

Avinash J. Agrawal received Bachelor of Engineering Degree in Computer Technology from Nagpur University, India, Master of Technology degree in Computer Technology from National Institute of Technology, Raipur, India and Ph.D. from Visvesvaraya National Institute of Technology, Nagpur, Indi in 1998, 2005 and 2013 respectively. His research area is Natural Language Processing and Databases. He is having 15 years of teaching experience. Presently he is Assistant Professor in Shri Ramdeobaba College of Engineering and Management, Nagpur. He is the author of several research papers in International and National Journal, Conferences.

How to cite this paper: Rohini P. Kamdi, Avinash J. Agrawal,"Keywords based Closed Domain Question Answering System for Indian Penal Code Sections and Indian Amendment Laws", International Journal of Intelligent Systems and Applications (IJISA), vol.7, no.12, pp.57-67, 2015. DOI: 10.5815/ijisa.2015.12.06