International Journal of Advance Research in Science and Engineering
Volume No.06, Special Issue No.(01), Nov 2017
www.ijarse.com
IJARSE
ISSN: 2319-8354

Automatic Question Generation System for Punjabi

Aarja Kaur[1] , Sukhpreet Singh[2]

[1] Assistant Professor , [2]Lecturer

[1,2] Department of Computer Application, Guru Kashi University, Talwandi Sabo,(India)

ABSTRACT

This paper introduces a form of Text-to-Question generation task, where the input text is in Gurumukhi script. The QG system would then generate a set of questions for which the sentence contains, implies, or needs answers. QG is actually a complex task, which relies on a large variety of resources and number of Natural Language Processing tools. This system relies on the Part of Speech Tagging (POS) tool. Also, various Punjabi Language dependent rules have been developed to generate output based on the POS Tags found in the given input sentence. Part of speech tagger that is applied on text to encode necessary information is used. It classify the sentences based on their subject, verb, object and preposition for determining the possible type of questions to be generated.

Keywords-Natural Language processing (NLP), Part of Speech Tagging (POS), Question Generation (QG)

I. INTRODUCTION

Questions are an important component of many educational interactions, from one-on-one tutoring sessions to large-scale assessments. Thus, the automation of generating questions would enable the ancient development of flexible and adaptive instructional technologies. For example, a teacher might use a question generation (QG) system to quickly create assessments for daily reading assignments. He or she might use an automated system to generate comprehension questions about a text found on the web, facilitating the use of practice reading texts that closely match students' interests. In the context of automated assessment, generating questions automatically from educational resources is a great challenge, with, potentially, tremendous impact. Computer systems can be used to generate questions automatically by writing algorithms for it. Thus, a Question Generation system can be designed to take input from the resources and generate a set of possible questions from the given input. Automated Question Generation (AQG) is a challenging task which involves natural language understanding and generation. Following are some applications of QG system.

- QG System helps learners to identify their knowledge deficits by asking questions. QG can also ask learners questions based on learning materials so that they can check their accomplishments and can focus on the keystones in study. The system can also help tutors in preparing question papers for the students to evaluate how well the students learned.

- QG system can also help in developing closed-domain Question Answering systems.
- QG system help in medicine by generating suggested questions for patients and caretakers.
- QG system helps to generate suggested questions that might be asked in security contexts by interrogators or in legal contexts by petitioners.
- QG can help to generate facilities like Frequently Asked Questions (FAQ) in which a list of FAQ is provided from a given information source. FAQ are the questions which are usually asked by the clients.

II.QUESTION GENERATION SYSTEM

In this section, detailed question generation system is described. The input is in Gurumukhi script and the output consists of possible questions which can be generated by the system for the given input. The question generation system proceeds by transforming declarative sentences into a set of possible questions. Garg and Goyal [8] has developed the System for Generating Questions Automatically from Given Punjabi Text by using Named Entity Recognition (NER) tool for the transformations. It uses various named entity lists to transform the declarative sentences into a set of questions. In current system Part of speech tagger that is applied on text to encode necessary information is used in this system. The sentences are

classified based on their subject, verb, object, preposition and other various tags for determining the possible type of questions to be generated. Tags used in system are Common Noun (N_NN), Personal Noun (N_NNP), NLoc (N_NST), Personal Pronoun (PR_PRP), Reflexive Pronoun (PR_PRF), Diectic Demonstrative (DM_DMD), Main Verb (V_VM), Nonfinite Verb (V_VM_VNF), Infinite Verb (V_VM_VINF), Auxiliary Verb (V_VAUX), Postposition (PSP), Adjective (JJ), Coordinator conjuction (CC_CCD), Default particles (RP_RPD), General quantifier (QT_QTF), Cardinal Quantifier (QT_QTC), Ordinal Quantifier (QT_QTO), Punctuation (RD_PUNC), the tag set defined by the *"Unified Parts of Speech (POS) Standard in Indian Languages by Department of Information Technology Ministry of Communications & Information Technology Govt. of India*

Copyright@TDIL." [12].

Punjabi language dependent rules have been developed to generate questions. By following these rules, for the given answer phrases, the system mainly tries to generate shallow questions with

question words: ਕੀ (*kī* (in Roman), What (English Meaning), ਕਦੋਂ(*kadō*ṃ, When), ਮਿਥੇ ਥੇ(*kitth*ē, Where),

ਮਕਸ (*kis*, Who/Whom), ਮਕੰ ਨੇ(*kinn*ē, How much/How many) etc.

A. Rules

Some Punjabi language dependent rules have been developed for the generation of questions. Some of these rules are:

International Journal of Advance Research in Science and Engineering
Volume No.06, Special Issue No.(01), Nov 2017
www.ijarse.com
IJARSE
ISSN: 2319-8354

Rule 1	*If there is input having 'N_NNP-N_NN- N_NNP' followed by 'PSP' then question is generated by substituting "ਮਕਸ" (kis)on the place of 'N_NNP-N_NN-N_NNP'.*

Example:

Input: ਸ਼ਹੀਦ ਭਗਤ ਮਸੰ ਘਦਾ ਜਨਿ ਮਦਹਾੜਾ ਿਨਾਇਆ| (shahīd bhagat siṅgh dā janam dihāṛā manāiā |)

Tagged data: ਸ਼ਹੀਦ\N_NNP ਭਗਤ\N_NN

ਮਸੰ ਘ\N_NNPਦਾ\PSP ਜਨਿ\N_NN ਮਦਹਾੜਾ\N_NN

ਿਨਾਇਆ\V_VM_VF|\RD_PUNC)

(shahīd\N_NNP bhagat\N_NN siṅgh\N_NNP dā\PSP janam\N_NN dihāṛā\N_NN manāiā\V_VM_VF |\RD_PUNC)

Output: ਮਕਸ ਦਾ ਜਨਿ ਮਦਹਾੜਾ ਿਨਾਇਆ? (kis dā janam dihāṛā manāiā?)

Rule 2	*If there is input having 'N_NNP-RD_PUNC-N_NNP-RD_PUNC-N_NNP-CC_CCD-N_NNP' followed by 'PSP then question is generated by substituting "ਮਕਸ-ਮਕਸ" (kis –kis) on the place of'N_NNP-RD_PUNC-N_NNP-RD_PUNC-N_NNP-CC_CCD-N_NNP.*

Example:

Input: ਮਕਿਸ਼ਨ, ਧਰਿਰਾਜ , ਯੁਮਧਸ਼ਟਰਅਤੇਅਰਜੁਨਦੇਿਨ

ਮਿਚ ਯੁਿੱਦੀਧ ਮਭਆਨਕਤਾ ਅਤੇਨਰਸੰ ਹਾਰਨ ੰਿੇਖਕੇ

ਭਾਰੀ ਮਗਲਾਨੀ ਹੋਈ|

(krishan , dharamrāj , yudhishṭar atē arjun dē man vic yuddh dī bhiānaktā atē narsaṃhār nūṃ vēkh kē bhārī gilānī hōī |)

Tagged data: ਮਕਿਸ਼ਨ\N_NNP ,\RD_PUNC

ਧਰਿਰਾਜ\N_NNP ,\RD_PUNC ਯੁਮਧਸ਼ਟਰ\N_NNP

ਅਤੇ\CC_CCD ਅਰਜੁਨ\N_NNPਦੇ\PSP ਿਨ\N_NN

ਮਿਚ\PSP ਯੁਿੱ\Nਧ_NN ਦੀ\PSP ਮਭਆਨਕਤਾ\N_NN

ਅਤੇ\CC_CCD ਨਰਸਂ ਹਾਰ\N_NN ਨ \PSPਂ ਿੇਖ\V_VM_VF

ਕੇ\CC_CCS ਭਾਰੀ\JJ ਮਗਲਾਨੀ\N_NN ਹੋਈ\V_VM_VNF

|\RD_PUNC)

(krishan\N_NNP ,\RD_PUNC dharamrāj\N_NNP ,\RD_PUNC yudhishṭar\N_NNP atē\CC_CCD arjun\N_NNPdē\PSP man\N_NN vic\PSP yuddh\N_NN dī\PSP bhiānaktā\N_NN atē\CC_CCD narsaṃhār\N_NN nūṃ\PSP vēkh\V_VM_VF kē\CC_CCS bhārī\JJ gilānī\N_NN hōī\V_VM_VNF |\RD_PUNC)

Output: ਮਕਸ –ਮਕਸ ਦੇਿਨ ਮਿਚ ਯੁਿੱਦੀਧ ਮਭਆਨਕਤਾ ਅਤੇ

ਨਰਸਂ ਹਾਰਨ ਂੰਿੇਖਕੇਭਾਰੀ ਮਗਲਾਨੀ ਹੋਈ?

(kis –kis dē man vic yuddh dī bhiānaktā atē narsaṃhār nūṃ vēkh kē bhārī gilānī hōī?)

Rule 3	*If there is input having 'N_NNP-RD_PUNC-N_NNP-RD_PUNC-N_NNP-CC_CCD-N_NNP' followed by 'V_VM_VNF' then question is generated by substituting "*ਮਕਿੱ-ਥੇਮਕਿੱ*"*ਥੇ *(kitthē-kitthē) on the place of 'N_NNP-RD_PUNC-N_NNP-RD_PUNC-N_NNP-CC_CCD-N_NNP'.*

International Journal of Advance Research in Science and Engineering
Volume No.06, Special Issue No.(01), Nov 2017
www.ijarse.com
IJARSE
ISSN: 2319-8354

Example:

Input: ਰੇਲਮਦਿੱ ਲੀ,ਪੁਣੇਅਤੇਨੇ ਇਡਾਜਾਂਦੀ ਹੈ| (rēl dillī, puṇē atē nōiḍā jāndī hai |)

Tagged data: ਰੇਲ\N_NN ਮਦਿੱ ਲੀ\N_NNP ,\RD_PUNC

ਪੁਣੇ\N_NNP ਅਤੇ\CC_CCD ਨੇ ਇਡਾ\N_NNP

ਜਾਂਦੀ\V_VM_VF ਹੈ\V_VAUX |\RD_PUNC)

(rēl\N_NN dillī\N_NNP,\RD_PUNC puṇē\N_NNP atē\CC_CCD nōiḍā\N_NNP jāndī\V_VM_VF hai\V_VAUX |\RD_PUNC)

Output: ਰੇਲਮਕਿੱ-ਥੇਮਕਿੱ ਥੇਜਾਂਦੀ ਹੈ? (rēl kitthē- kitthē jāndī hai?)

Rule 4	*If there is input having 'N_NNP' followed by 'V_VM_VNF 'then question is generated by substituting "*ਮਕਿੱ"ਥੇ *(kitthē) on the place of 'N_NNP'.*

Example:

Input: ਿਿੱਧਰੇਲਿਦੀ ਿੁੰਬਈ_ ਭੋਪਾਲ ਮਦਿੱ ਲੀਲਾਈਨ ਦੇ

ਭੋਪਾਲਸਟੇਸ਼ਨਤੋਂਇਕ ਲਾਈਨ ਉਜੈਨਜਾਂਦੀ ਹੈ|

(maddh rēlvē dī mumbī _ bhōpāl_dillī lāīn dē bhōpāl saṭēshan tōṃ ik lāīn ujain jāndī hai |)

Tagged data: ਿਿੱਧ\NN_NN ਰੇਲਿ\N_NN ਦੀ\PSP

ਿੁੰਬਈ\N_NNP _\RD_SYM ਭੋਪਾਲ ਮਦਿੱ ਲੀ\N_NNP

ਲਾਈਨ\N_NN ਦੇ\PSP ਭੋਪਾਲ\N_NNP ਸਟੇਸ਼ਨ\N_NN

ਤੋਂ\PSP ਇਕ\QT_QTC ਲਾਈਨ\N_NN ਉਜੈਨ\N_NNP

ਜਾਂਦੀ\V_VM_VF ਹੈ\V_VAUX |\RD_PUNC)

(maddh\N_NN rēlvē\N_NN dī\PSP mumbī\N_NNP _\RD_SYM bhōpāl_dillī\N_NNP lāīn\N_NN dē\PSP bhōpāl\N_NNP saṭēshan\N_NN tōṃ\PSP ik\QT_QTC lāīn\N_NN ujain\N_NNP jāndī\V_VM_VF hai\V_VAUX |\RD_PUNC)

Output: ਿਿੱਧਰੇਲਿਦੀ ਿੁਂਬਈ_ਭੋਪਾਲ ਮਦਿੱਲੀਲਾਈਨ ਦੇ

ਭੋਪਾਲਸਟੇਸ਼ਨਤੋਂਇਕ ਲਾਈਨ ਮਕਿੱਥੇਜਾਂਦੀ ਹੈ?

(maddh rēlvē dī mumbī _bhōpāl_dillī lāīn dē bhōpāl saṭēshan tōṃ ik lāīn kitthē jāndī hai?)

III. ARCHITECTURE.

Architecture of the system is divided into five phases

1. Split the set of input data into number of sentences by dandi '|' (puran viram) at end of each sentence.
2. Get POS tag for each word of sentence.
3. Fetch the rules from Database and apply the number of suitable rules on input sentence.
4. Post-Processing to ensure proper formatting.
5. Repeat the process for next sentence.

IV. EVALUATION

To evaluate the performance of our question generation system, we have used three standard mertics namely Precision, Recall and F-measure.The analysis of output generated by POS Tagging based QG system has been done manually. We have collected test data from various Punjabi websites and Punjabi text books.

The system has been tested for a test data of 400 sentences. In the test data, there were mostly sentences for which multiple questions are generated by the system. So, from these 400 sentences, the system could generate 1220 questions and manually we could generate 1650. So, the overall recall of the system for the test data has been calculated to 73.93%.

For evaluating precision, grammatically wrong constructed questions have not been considered as the valid questions. The system was able to generate 950 grammatically correct questions from 1220 generated questions. The overall precision of the system for the test data has been calculated to 77.8 %. Tables1&2 shows the individual recall, precision and F-measure values obtained for the various types of questions, respectively.

International Journal of Advance Research in Science and Engineering
Volume No.06, Special Issue No.(01), Nov 2017
www.ijarse.com
IJARSE
ISSN: 2319-8354

Table 1. Results with Question Type and Recall %

Question Type	Recall %
ਮਕਸ,ਮਕਸਦਾ,ਮਕਸਦੀ,ਮਕਸਦੇ,ਮਕਸਨ ◌ੰ (Whom)	55.7
ਮਕਿੱ ਥੇ(Where)	48.0
ਕੀ (What)	59.2
ਕਦੋਂ(When)	68.0
ਮਕੰ ਨ,◌ੇਮਕੰ ◌ਿੀ,ਮਕੰ ◌ਿਾ,ਮਕੰ ◌ਿੇ (How many / How much)	72.6
ਮਕਹੜਾ,ਮਕਹੜੇ(Which)	52.0
ਕੋਣ(who)	26.6

Table 2. Results with Question Type and Precision %

Question Type	Precision%
ਮਕਸ,ਮਕਸਦਾ,ਮਕਸਦੀ,ਮਕਸਦੇ,ਮਕਸਨ ◌ੰ (Whom)	73.1
ਮਕਿੱ ਥੇ(Where)	66.6
ਕੀ (What)	67.7
ਕਦੋਂ(When)	41.1
ਮਕੰ ਨੇ,ਮਕੰ ◌ਿੀ,ਮਕੰ ◌ਿਾ,ਮਕੰ ◌ਿੇ (How many / How much)	68.8
ਮਕਹੜਾ,ਮਕਹੜੇ(Which)	70.2
ਕੋਣ (who)	75.0

V. CONCLUSION AND FUTURE SCOPE

Work on Question Generation System for Punjabi Language have been reported. A number of language dependent rules are formed to extract language dependent features for Punjabi.The system shows good results for number of question types but for some question types the system shows low results. So, future work includes forming new rules to improve the existing rules.As person and location both comes under proper nouns, this limitation lowers the performance of the system. This issue can be considered to improve the system. From the possible Answer Phrase, the system generate

questions with the question words ਮਕਸ(kis),

ਮਕਸਦਾ(kisadā), ਮਕਸਦੀ(kisadī), ਮਕਸਦੇ(kisdē),

ਮਕਸਨ (ੰkisnūṃ); □□□□(□□kitthē); □□(kī); □□□□(kadōṃ);

□□(kiṃvē);

□□□□□(kihṛā), □□□□□(kihṛē); □□□(kōṇ). In future, the

system could be extended to detect and transform other types of phrases to produce other types of

questions like □□□□ (kiuṃ), □□□□□(kivēṃ).

In case of multiple sentences in the input text, the system does not consider any relationship between the different sentences. By taking into consideration the relationships between sentences, the system can give better results for the multiple sentences text. As system is using POS tagger or POS tagged data. So it difficult to make rules for every relation in words. So, in future more POS tags can be used to improve the performance of the system. To develop the QG system for Punjabi Language more NLP tools are used for getting more accuracy and more generated questions. In future other essential tools for QG for Punjabi Language can be developed.

REFERENCES

[1]Dieguez D, Rodrigues R, Gomes P Using CBR for Portuguese Question Generation. In: *Proceedings of the 15th Portuguese Conference on Artificial Intelligence*, 2011, 328- 341.

[2]Heilman M,Automatic Factual Question Generation from Text. PhD Dissertation, Carnegie Mellon University,2011

[3]Ali H, Chali Y, Hasan SA (2010) Automation of Question Generation from Sentences, *Proceedings of QG2010: The Third Workshop on Question Generation*, 58-67.

[4]Mannem P, Prasad R, Joshi A,Question Generation from Paragraphs at UPenn: *QGSTEC System Description. In: Proceedings of QG*,2010, 84-91.

[5]Heilman M, Smith NA (2010) Good Question! Statistical Ranking for Question Generation. In: Human Language Technologies: *The 2010 Annual Conference of the North American Chapter of the Association for*

Computational Linguistics, 609-617.

[6]Bernhard D, De Viron L, Moriceau V, Tannier X Question Generation for French: *Collating Parsers and Paraphrasing. Dialogue & Discourse 3*, 2012, 43-74.

[7]Curto S, Mendes AC, Coheur L, *Question Generation based on Lexico-Syntactic Patterns Learned from the Web. Dialogue & Discourse 3*, 2012 147-175.

[8]Garg S, Goyal V,Automatic Question Generation for Punjabi, *International Journal of Computer Science and Engineering* ,2012,324-327.

[9]Kalady S, Elikkottil A, Das R, Natural Language Question Generation Using Syntax and Keywords, *In Proceedings of QG* , 2010, 1-10.

*[10]*Piwek P, Boyer KE ,Varieties of Question Generation: *Introduction to this Special Issue. Dialogue and Discourse 3*, 2012,1-9.

[11] Kumar V (2013) A Question Generator System Using Stanford Parsing,*International Journal of Engineering Research and Development 7*,1-5