Domain specific Question & Answer generation in Tamil

Rubika Murugathas
Department of Computer Science Engineering
University of Moratuwa
Colombo, Sri Lanka
rubika.murugathas.19@cse.mrt.ac.lk

Uthayasanker Thayasivam
Department of Computer Science Engineering
University of Moratuwa
Colombo, Sri Lanka
rtuthaya@cse.mrt.ac.lk

***Abstract*— Automatic Question & Answer generation is a challenging task in natural language processing. The proposed system is capable of automatically generating questions and answers from a given history related text content in Tamil. It processes the input text using various NLP techniques. The system has four modules namely, Preprocessing module, Rule-based module, Named Entity Recognition (NER) module, Question Answer Generator (QAG) module. Regex patterns and gazetteers are used in rule-based module and machine learning approach is used for NER module. A NER module built using Conditional Random Field (CRF) classifier is used which is trained on a manually tagged dataset for history domain in Tamil. Questions are formed using grammatical and defined rules from the named entities identified from both rule-based and NER module. An affix stripping algorithm implemented to find the inflection suffix. A history text from Wikipedia is evaluated by 16 native Tamil speakers under categories like undergraduates, graduates and experts. According to the evaluation results, 62.22% of total generated questions are grammatically correct and meaningful questions despite the domain and language related challenges.**

Keywords*— *question and answer generation, tamil, NER, CRF, history, domain specific, QAG

I. Introduction

E-learning is widely used in many countries and it help students to learn on their own by accessing materials from internet. Question and Answer generating system from a given text helps the students to learn more efficiently and evaluate themselves. Practicing more questions will make students more prepared for the competitive exams and improve their knowledge in answering questions in exams. It also helps the examiners to prepare questions for exams.

Automatic QA generators are developed using syntax based, semantic based and template based approaches. Syntax based approaches focus on the syntax such as POS, NER, syntactic tree, etc. Semantic based approaches focus on a deeper level and mostly uses a knowledge source like taxonomies, Ontologies, etc. Template based approaches use templates that define structure of question using fixed text and placeholders [12]. Rules, templates or statistical methods are used for question construction. Rules are used to select suitable question type and to construct questions. Statistical methods learns question transformation from training data. Recently neural techniques are used to generate questions. Most of the researches that uses neural techniques follow the sequence-to-sequence framework.

There are systems and online applications available for question banks with marking schemes. Also few AI applications and libraries available in internet to generate questions with answers from a given input text in English. Question Answering system and Question and Answer generation system are two different researches where the first finds the answer from given text when a question is provided, while the second generates possible questions and answers both from a given text. There are more researches conducted in Question Answering system compared to Question and Answer generation in Tamil. While there are number of researches conducted on developing various NLP tools in Tamil, only very few libraries and tools are publicly available. There are only very few researches carried out in generating question and answers in Tamil using the available or developed toolkits.

The proposed system is an automatic Question and Answer Generator (QAG), developed as a first attempt for history domain in Tamil language. The system is intended to help the students studying in Tamil medium to learn more efficiently and evaluate themselves for history subject. It also benefits the examiners in preparing quizzes and exam papers using the system. The system accepts a history related textual content in Tamil language and processes the text paragraph sentence by sentence to generate possible WH questions with answers. The system generates simple factoid questions, but it is not capable of generating complex questions like open or discussion questions

The system uses a Rule-based module as well as Named Entity Recognizer (NER) module for generating questions. Regex patterns and gazetteers are used for the implementation of rule based module. NER is built using machine learning technologies using a manually tagged dataset as there are no dataset publicly available for history domain in Tamil. The questions generated from the final system is evaluated on the grammatical and semantical correctness using crowdsourcing review from a number of category of people including experts. The system performs well in producing more meaningful questions despite the domain and language related challenges.

II. Related Work

A. Automatic QA generation in Tamil

Vignesh N and S.Sowmya [2] proposed a system to automatically generate questions in Tamil from a given input text. Words in the sentence are tagged with Noun, Verb, Time, Noun descriptor and Verb descriptor tags. Verb tags have sub tags gender and tense. Gender, tense, Noun descriptor and verb descriptor are tagged based on case markers. After tagging, the question word is replaced in the place of descriptor tags using some defined rules.

B. Automatic QA generation in other languages

Deepali et.al [6] developed a rule based question generation for Marathi text summarization. POS tagging applied and nouns are further classified as person and

location using NER tools. Affix stripping algorithms are used for rule based stemming. Holy Lovenia et.al [4] presented a rule-based automatic QA system for reading comprehension. Sentence selection done using text summarization method Text rank and Latent Semantic Analysis (LSA). The gap selection is done using constituent parsing and Named Entity Recognition. NER was experimented with CRF and Bi-LSTM CRF on generic tags. CRF outperformed deep learning model. The fill in the blank question answer pair is converted to WH questions by converting the sentence to interrogative form. The question word is determined by the named entity type of its answer. Dhaval et.al. [7] developed a rule based QA generation system to generate simple and complex type of questions. Sentence selection is done by feature extraction such as number of nouns and pronouns, length, etc. To generate simple questions, Named Entity Recognizer is used. Complex questions are formed by identifying discourse connective words such as because, since, as a result, etc.

Du et al. [13] proposed the first sentence level sequence tagging model using neural techniques. The encoder uses a BiLSTM and an attention mechanism to help decoder focus on most relevant parts of the sentence and a decoder using LSTM to generate question. Seq2Seq models struggle to utilize relevant contexts avoiding irrelevant information [22]. Zhao et al. [15] proposes a maxout pointer mechanism with gated self-attention encoder to address the problems of processing long text in previous sequence neural models and achieved a new state-of-art results for the SQUAD dataset. Kim et al. [16] proposed a novel answer-seperated seq2seq architecture with 2 encoders separately for paragraph and answer and answer-seperated decoder. A keyword-net used to obtain the key information from answer. Contextual feature of paragraph from attention mechanism and keyword feature from keyword-net is used by decoder to generate a question.

Fabbri et al. [17] proposed a template-based model that uses a retrieval based approach to obtain a sentence from the corpus identical to the current context. A pretrained BERT model was fine-tuned on the data created with question for all context & answer pairs and was evaluated on the SQUAD dataset. Teshani et al.[5] developed a QA Generator for Sinhala using a semantic relationship identifier that defines 8 patterns for Sinhala sentence with the basic subject, object, and verb with pronouns, adjectives, adverbs.

Sathish et al. [19] proposed an automatic Question Answer pair generator using a knowledge graph. A set of keywords from entities and relationship stored as subject, a predicate and object triple in knowledge graph. A sequence to sequence model using RNN is built using subset of keywords as sequence. Geetanjali et al.[20] developed a QA generator by mapping Abstract Meaning representation (AMR) to question-answer meaning representation (QMR). AMR is a semantic representation of a whole sentence and templates are defined for the relations in AMR. Templates are transformed to suitable questions.

C. *Evaluation of QA generation*

The most common evaluation approach is expert evaluation. The second most common approach is comparing machine generated questions with human authored questions [12]. BLEU [13, 14, 16, 18, 19], METEOR [13, 16, 18] and ROUGE [14, 16] are some popular metrics that compute the n-gram similarity between the reference sentence and the generated sentence [23] and used by most of the neural QA generator models. Jouault et al. [9] evaluated the coverage of questions generated by expert from the same text is also evaluated. Mazidi et al.[10] has evaluated the question acceptability using 5 point scale using crowdsourcing review evaluation method. Zhang et al. [11] has evaluated the semantic ambiguity of questions generated using 5 point scale review from 12 students. Huang and He [8] evaluated the question acceptability using binary scale from 4 experts and difficulty and discrimination is evaluated by conducting mock exam for students. Flor & Riordan [21] evaluated the semantical and grammatical correctness with 5 point scale review from 2 experts.

III. METHODOLOGY

The system is capable of generating possible questions and answers on history related text in Tamil language. It is mainly composed of 4 components namely Preprocessing module, Rule based module, Named Entity Recognition module and Question and Answer generation module. Fig.1. depicts the high level diagram of the proposed system.

A. *Preprocessing Module*

Text pre-processing is a process of transforming text into a more digestible form to be used for further processing. Below are the steps followed in preprocessing.

- Unicode conversion – Convert different tamil encodings to Unicode.
- Tokenization - The text content is tokenized into list of sentences.
- Sentence selection - The sentences suitable for question generation are selected by checking the first word of the sentence against a defined list of anaphora word list. Anaphora is the linguistic phenomenon for referring back to a noun or pronoun in previous sentences.

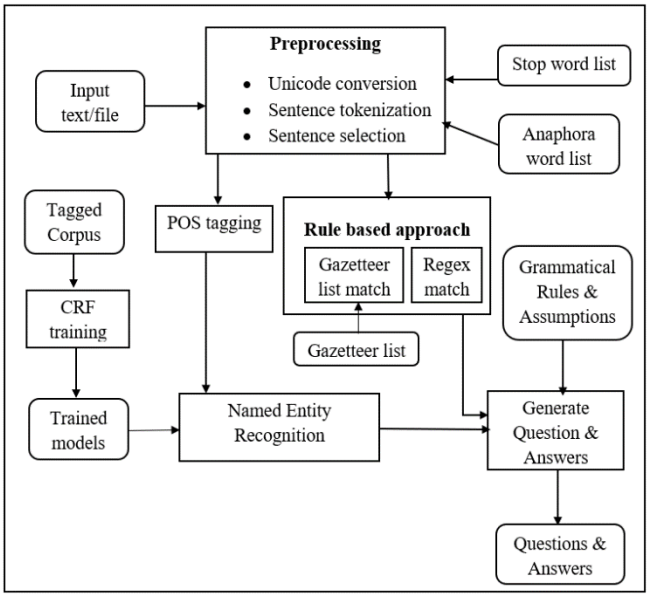

Fig.1.High level diagram of QAG system

B. Rule-based module

a) Regex Match : Regex match is checked to identify date, time and quantity in the sentence. The matching word is replaced with the appropriate question word to form questions of type 'when' and 'how many'. Regex pattern for comma separated words of same entity type, words enclosed with single quote are also checked against sentence and is replaced with blank line if match found.

b) Gazetteer Match : A gazetteer list for few named entities are defined. The words in the sentence is checked against this list and if a word is found in an array the corresponding key is returned. According to the returned key, the question word is replaced.

- Month - List of 12 months
- Kingdom - List of all Srilankan kingdoms
- City - List of historic cities and capital cities
- Person - List of famous kings, emperors, rulers
- Country - List of historically important countries.
- Numeric word - List of numeric words under categories of units, tens, hundred, thousand scales.

C. Named Entity Recognition module

Named entity recognition (NER) is used to classify named entities in text into pre-defined categories such as the names of person, organization, location, date, time, quantities, etc.[3]. A NER system developed by Rubika et al. [1] for history domain in Tamil is used for the proposed system. Machine learning approach is used and novel entity tag set for history domain is trained. Conditional Random Field (CRF) classifier is applied with hyper parameter optimization. Different features suitable for the domain of interest and language like context word, inflection suffix, clue words, POS tag, gazetteer, stem word, prefix suffix rule, etc. are used. Random search algorithm is used for hyper parameter tuning to get the highest performing model. Named entity tag set contains 36 named entity tags like person, kingdom, country, city, law, antiquity, equipment, government, troop, event, etc. Most of them are domain specific [1].

D. Question and Answer generation module

The named entities identified by both rule-based module and NER module follows the below steps to generate questions and answers.

1. The sentence, identified named entity, named entity type are given as input.
2. The previous and next words of identified named entity are stemmed.
3. The stemmed words checked against clue word list.
4. If previous or next words are clue words, then the appropriate question word is replaced with inflection suffix in the place of word and previous or next word which contains the clue word.
   Sentence: 1914 முதல் 1918 வரை நடந்த முதலாவது உலக மகாயுத்தம் ஜேர்மனி நாட்டின் தலைமையிலான அச்சு நாடுகளின் தோல்வியுடன் முடிவுக்கு வந்தது.
   The First World War which lasted from 1914 to 1918 ended with the defeat of the Axis Powers led by country of Germany.
   Question: 1914 முதல் 1918 வரை நடந்த முதலாவது உலக மகாயுத்தம் எந்த நாட்டின் தலைமையிலான அச்சு நாடுகளின் தோல்வியுடன் முடிவுக்கு வந்தது?
   The First World War which lasted from 1914 to 1918 ended with the defeat of the Axis powers led by which country?

   The word 'ஜேர்மனி' (Germany) is identified as a country named entity and it follows the word 'நாட்டின்'(of the country) which is inflected in Tamil. The stem word of the following word is 'நாடு'(country) and it is present in the clue word list of named entity 'country'. Hence the identified named entity word with its following word is replaced with a question word.
   'ஜேர்மனி நாட்டின்' (country of Germany) is replaced with the question word 'எந்த நாட்டின்' (by which country)

5. If previous or next words are not clue words, then the appropriate question word is replaced with inflection suffix in the place of identified named entity word.
   Example:
   Sentence: 1914 ஆகஸ்ட் முதலாம் திகதி ஜேர்மனி ரஷ்யாவுக்கெதிராக யுத்தப் பிரகடனம் செய்தது. (In 1914 August 1st, Germany declared a war against Russia.)
   Question: 1914 ஆகஸ்ட் முதலாம் திகதி எந்த நாடு ரஷ்யாவுக்கெதிராக யுத்தப் பிரகடனம் செய்தது? (Which country declared war against Russia in 1st of August 1914?)
   The word 'ஜேர்மனி' (Germany) is identified as a country named entity and it does not proceed or precede any clue words. Hence the named entity word is replaced with appropriate question word 'எந்த நாடு' (Which country).

a) Rules and Assumptions

- The named entities of type Government, Kingdom, Organization, Troop, Law, Water bodies, Tax, Industry, Sector are followed by clue words most of the cases. Hence these named entities are replaced with question word if only clue words are present in the predicted entity.
- A clue word proceeding a named entity has a limited possible inflection suffix. In the system, only if the previous word containing clue word with inflection suffix either of 'ஆன' or 'ஆகிய' are considered while replacing question word.
- If a named entity ending with comma and the following word is of same named entity type then question formation for that named entity type is ignored in NER module and handled in regex pattern approach.
- If a named entity followed by a determiner words like 'என்ற', 'என்கின்ற' most probably it follows with the clue word of the named entity. So one word after the determiner is analyzed when replacing question word.

b) Pseudocode

Input: sentence, named entity (W[i]), previous word (W[i-1]), next word (W[i+1]), named entity type

Process:

IF *part word of W[i] or stem word is clue word*
Except the part word, other tokens replaced with question word 'எந்த'
IF *W[i-1] or stem word is a clue word*
Get the inflection suffix of W[i-1]
IF *inflection suffix is 'ஆகிய' or 'ஆன'*
Check previous word W[i-2]
IF *ends with rhyme 'அ'*
Get inflection suffix of W[i]
Get question word for NE type & inflection of W[i]
Replace W[i-2],W[i-1], W[i] with question word
Write question and answer to file
Return
ELSE
Get inflection suffix of W[i]
Get question word for NE type & inflection of W[i]
Replace W[i-1], W[i] with question word
Write question and answer to file
Return

ELSE IF *W[i+1] or stem word has clue word*
Replace W[i] with question word 'எந்த'
Write question and answer to file
Return

ELSE IF *W[i+1] is a word in ('என்ற', 'எனப்படும்' , 'என்கின்ற') and W[i+2] is a clue word*
Replace W[i], W[i+1] with question word 'எந்த'
Write question and answer to file
Return

ELSE IF *W[i] is a PER entity and W[i+1] or stem word of W[i+1] is a word in ('என்பவர்', 'என்கின்றவர்')*
Replace W[i], W[i+1] with question word 'யார்'
Write question and answer to file
Return

ELSE IF *no clue words present in previous, next or current word and named entity not in ['LAW', 'GOV', 'ORG', 'TRO', 'IND', 'SEC', 'SKILL', 'SOU', 'TAX', 'LOC']*
Get inflection suffix of W[i]
Get question word for NE type & inflection of W[i]
Replace W[i] with question word
Write question and answer to file
Return

c) Affix stripping : An affix stripping algorithm is implemented to extract the inflection suffix from a word. Tamil characters can have 2 Unicode points for a single character. It needs to be extracted in order to find the inflection suffix.

யி => [ய், இ]
வி => [வ், இ]

Inflection suffix identified by algorithm: ஐ, ஆல், கு, இல், இன், அது, உடன், உடைய, மீது, ஆகிய, இருந்து, உம், இலும், ஆன, இடம், ஆக, ஆம், ஆவது.

IV. Experiments And Evaluation

Questions are evaluated for syntactic and semantic correctness manually and error analysis is done on the evaluation results. The system is improved with few post processing rules and a final evaluation is done on the improved system. The system has few limitations as well.

a) Post processing rules

- If identified named entity ends with comma and does not have an inflection suffix then gap fill question is formed be replacing named entity with blank line.
- If named entity proceeded by word 'என', 'என்று' the question formation is ignored.
- If the numeric words are preceded by words like பல (several) and சில (few), then 'how many' type of question formation is ignored.
- If a numeric entity is followed by list of comma seperated words, the question formation can be ignored as it may indicate the number of categories which can be predicted from the question itself.

b) Limitations

- The system cannot identify if subject is missing in the sentence.
- The system is not capable of identifying if context is missing in the sentence.
- When non inflected words ends with any inflection suffix, the system will not be able to distinguish.
- If anopharic words occur in middle of sentence, the sentence is not ignored for question generation.
- After a sentence replaced with question word, it can contain the answer or clue for an answer in the question itself which can produce inappropriate questions.

c) Evaluation : A history related text content from Wikipedia Tamil is used for evaluation. The evaluation is done from 16 native Tamil speakers from various categories like undergraduates, experts (History teachers) and others (graduates and employees). Out of 16 evaluators 3 are experts. All 16 evaluators have done History as subject in their secondary or higher education. The questions are evaluated on the correctness of question format (correct question word and inflection suffix of the question word) and the appropriateness of questions (if context of question is present and question is meaningful). Below 4 options are given to evaluate each of 44 questions and answers generated by the system.

Option 1: கேள்வி வடிவம் தவறானது மற்றும் கேள்வி அர்த்தமற்றது. (Question format is wrong and question is meaningless)

Option 2: கேள்வி வடிவம் சரியானது, ஆனால் கேள்வி பொருத்தமற்றது. (Question format is correct, but question is not appropriate)

Option 3: கேள்வி வடிவம் தவறானது. எனினும் கேள்வி அர்த்தமுள்ளது. (Question format is wrong, but still the question is meaningful)

Option 4: கேள்வி வடிவம் சரியானது, மற்றும் கேள்வி பொருத்தமானது. (Question format is correct and question is appropriate)

TABLE.I. PERFORMANCR METRICS FOR QA GENERATION

Questions generated	NER Module	Rule-based Module	
		Regex Match	*Gazetteers*
Total questions generated	38.63%	50%	11.36%
Meaningful questions	71.27%	88.1%	86.3%
Grammatically correct questions	69.65%	76.82%	66.28%
Both grammatically correct and Meaningful questions	55.9%	69.74%	62.52%

Option 2 & 4 indicates grammatically correct questions and Option 3 & 4 indicates meaningful questions. Fig.2. & Fig.3. illustrates the percentage of meaningful and grammatical correct questions generated by system calculated from the average of responses from all 16 evaluators & experts respectively. Table I indicates the performance metrics of questions generated from rule-based and NER module separately.

d) Inter-rater reliability : Krippendorff's Alpha (α) is the inter-rater reliability coefficient used to measure the agreement among the 16 raters. Estimates calculated for both grammatical & semantical correctness evaluation separately using SPSS tool is depicted in Table II. It achieved an alpha value of 0.5750 & 0.4426 respectively which shows a moderate agreement between raters.

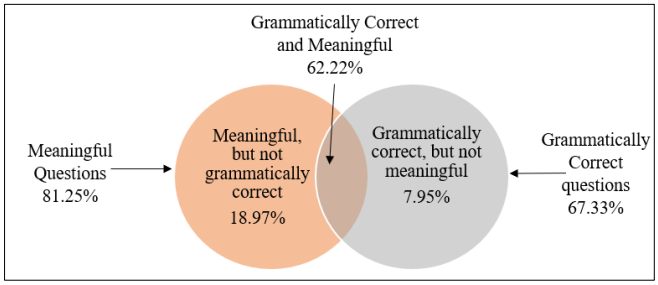

Fig.2. Question Evaluation Results from all evaluators

TABLE.II. INTER-RATER RELIABILITY ESTIMATE FOR QA EVALUATION

Metrics		Grammatical Correctness	Semantical Correctness
Alpha		**0.5750**	**0.4426**
LL95%CI		0.4599	0.3033
UL95%CI		0.6812	0.5720
Units		44.0000	44.0000
Observers		16.0000	16.0000
Pairs		5280.0000	5280.0000
Number of bootstrap samples		10000	10000
Probability of failure to achieve an alpha of at least alphamin	.9000	1.0000	1.0000
	.8000	1.0000	1.0000
	.7000	0.9921	1.0000
	.6700	0.9587	0.9996
	.6000	0.6530	0.9889
	.5000	0.0939	0.7706

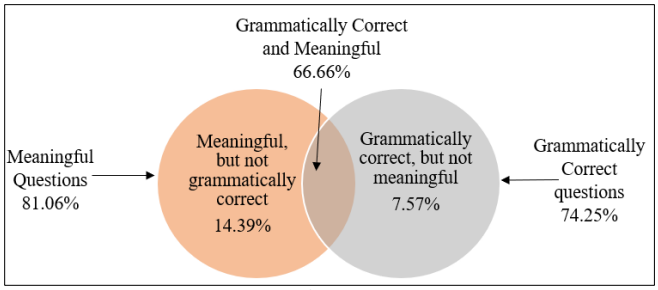

Fig.3. Question Evaluation Results from experts

e) Results: Half of the questions generated are from regex match approach in rule based module. More than 85% of questions generated by rule based module are meaningful. While around 71% of questions generated by NER module are meaningful as it identifies the named entity wrong or identifies partially in some cases. Most of the questions generated by NER module is 'which country' type questions as NER module identifies country tag correctly in most of the cases due to the fact that dataset has more country tags. Around 65-70% of the questions generated by the NER module and gazetteer approach are grammatically correct.

V. Analysis

Tamil is a low resource languages with limited dataset, NLP libraries and tools that are publicly available. There are very few researches done on QA generation in Tamil and there is no research done so far for QA generation for history domain in Tamil.

The proposed system is the first attempt for QA generation for history domain in Tamil language. The root word of next and previous words are checked to ensure any clue words exist before forming questions. But when the root word is not identified correctly, it produces a question with incorrect format. The sentence that has anaphora words in the middle are not identified during sentence selection. Also the sentence without subject or object is possible in Tamil language. These types of sentences are not ignored. Hence it can produce meaningless questions even if the question formed is grammatically correct. In few case, only part of a multi-word named is correctly identified. So when replacing with the question word this will generate wrong format question. When extracting inflection suffix using affix stripping, there is a possibility a word without inflection get stripped in few cases. Hence this could also produce wrong format questions. The system performs well in producing more meaningful questions compared to grammatically correct questions according to the evaluation results.

VI. Conclusion & Future Work

A rule-based Q&A generator capable of generating questions and answers from a given history related textual content in Tamil is presented. The system generates factoid WH questions and gap fill questions from a Rule-based module & Named Entity Recognition (NER) module. Regex patterns and gazetteers list are used in Rule-based module. A NER module built using a CRF classifier with hyper parameter tuning is used. The named entities identified from both modules is replaced with a suitable question word using defined rules. An affix stripping algorithm is

implemented to extract the inflection suffix. The system was manually evaluated and then post processing rules were applied based on the error analysis done. The generated questions and answers from the final improved system is evaluated by 16 native Tamil speakers from different categories including 3 experts. According to the evaluation results, 62.22% of total generated questions are both grammatically and semantically correct when considering the average of responses. Krippendorff's Alpha is used for the measure of inter-rater reliability that achieved a moderate agreement with values 0.5750 & 0.4426 respectively for grammatical & semantical correctness ratings.

NER module built with deep learning techniques could enhance the performance in identifying named entities. Text summarization can be performed before question generation to produce more meaningful questions. Anaphoric resolution to handle any word reference from previous sentence could be implemented to produce more and meaningful questions. Further the system can be enhanced to support different question types & domains.

REFERENCES

[1] Murugathas, Rubika and Thayasivam, Uthayasanker. "Domain specific Named Entity Recognizer in Tamil" in press.

[2] Vignesh N and S.Sowmya. "Automatic Question Generator in Tamil." International Journal of Engineering Research & Technology (IJERT) Vol. 2 Issue 10, October - 2013 IJERTIJERT ISSN: 2278-0181

[3] Prakash Hiremath & Shambhavi B. R. A Survey on Named Entity Recognition for South Indian Languages International Journal of Engineering and Advanced Technology (IJEAT) ISSN: 2249 – 8958, Volume-3 Issue-6, August 2014.

[4] Lovenia, Holy & Limanta, Felix & Gunawan, Agus. (2018). Automatic Question-Answer Pairs Generation from Text. 10.13140/RG.2.2.33776.92162.

[5] Dissanayake, Teshani & Hettige, Budditha. (2021). Thematic Relations Based QA Generator for Sinhala.

[6] Deepali K. Gaikwad , C. Namrata Mahender. 2018."Question Generation System for Marathi Text" International Journal of Scientific Research in Computer Science, Engineering and Information Technology (IJSRCSEIT).(2017).Vol.3(ISSN : 2456-3307)

[7] Dhaval Swali, Jay Palan, Ishita Shah. "Automatic Question Generation from Paragraph", International Journal of Advanced Engineering and Research Development.

[8] Huang, Y., & He, L. (2016). Automatic generation of short answer questions for reading comprehension assessment. Natural Language Engineering, 22(3), 457–489.

[9] Jouault, C., Seta, K., Hayashi, Y. (2017). SOLS: An LOD based semantically enhanced open learning space supporting self-directed learning of history. IEICE Transactions on Information and Systems, 100(10), 2556–2566

[10] Mazidi, K., & Tarau, P. (2016b). Infusing NLU into automatic question generation. In: the 9th International Natural Language Generation conference, pp. 51–60.

[11] Zhang, L., & VanLehn, K. (2016). How do machine-generated questions compare to human-generated questions? Research and Practice in Technology Enhanced Learning, 11(7).

[12] Kurdi, Ghader & Leo, Jared & Parsia, Bijan & Sattler, Uli & Al-Emari, Salam. (2019). A Systematic Review of Automatic Question Generation for Educational Purposes. International Journal of Artificial Intelligence in Education. 30. 10.1007/s40593-019-00186-y.

[13] Xinya Du and Claire Cardie. 2017. Identifying where to focus in reading comprehension for neural question generation. In Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 2067–2073.

[14] Sun, Yuan & Chen, Chaofan & Chen, Andong & Zhao, Xiaobing. (2021). Tibetan Question Generation Based on Sequence to Sequence Model. Computers, Materials & Continua. 68. 3203-3213. 10.32604/cmc.2021.016517.

[15] Yao Zhao, XiaochuanNi, YuanyuanDing, and Qifa Ke. 2018. Paragraph-level neural question generation with maxout pointer and gated self-attention networks. In Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 3901–3910.

[16] Yanghoon Kim, Hwanhee Lee, Joongbo Shin, and Kyomin Jung. 2019. Improving neural question generation using answer separation. In AAAI Conference on Artificial Intelligence (AAAI).

[17] Alexander Fabbri, Patrick Ng, Zhiguo Wang, Ramesh Nallapati, and Bing Xiang. 2020. Template-Based Question Generation from Retrieved Sentences for Improved Unsupervised Question Answering. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 4508–4513, Online. Association for Computational Linguistics.

[18] Iulian Vlad Serban, Alberto García-Durán, Caglar Gulcehre, Sungjin Ahn, Sarath Chandar, Aaron Courville, and Yoshua Bengio. 2016. Generating Factoid Questions With Recurrent Neural Networks: The 30M Factoid Question-Answer Corpus. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 588–598, Berlin, Germany. Association for Computational Linguistics.

[19] Reddy, Sathish & Raghu, Dinesh & Khapra, Mitesh & Joshi, Sachindra. (2017). Generating Natural Language Question-Answer Pairs from a Knowledge Graph Using a RNN Based Question Generation Model. 376-385. 10.18653/v1/E17-1036.

[20] Rakshit, Geetanjali & Flanigan, Jeffrey. (2021). ASQ: Automatically Generating Question-Answer Pairs using AMRs.

[21] Flor, M., & Riordan, B. (2018). A semantic role-based approach to open-domain automatic question generation. In: the 13th Workshop on Innovative Use of NLP for Building Educational Applications, pp. 254–263.

[22] Pan, Liangming & Lei, Wenqiang & Chua, Tat-Seng & Kan, Min-Yen. (2019). Recent Advances in Neural Question Generation.

[23] Preksha Nema and Mitesh M. Khapra. 2018. Towards a Better Metric for Evaluating Question Generation Systems. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 3950–3959, Brussels, Belgium. Association for Computational Linguistics.

[24] Blsť ak, M., & Rozinajov ´ a, V. (2017). Machine learning approach to the process of question generation. ´ In Blsť ak, M., & Rozinajov ´ a, V. (Eds.) ´ Text, speech, and dialogue (pp. 102–110). Cham: Springer International Publishing

[25] Kaur, A., & Singh, S. (2017). Automatic question generation system for Punjabi. In: The international conference on recent innovations in science, Agriculture, Engineering and Management

[26] Liu, M., Rus, V., Liu, L. (2017). Automatic Chinese factual question generation. IEEE Transactions on Learning Technologies, 10(2), 194–204.

[27] Blsť ak, M., & Rozinajov ´ a, V. (2018). Building an agent for factual question generation task. In ´ 2018 World symposium on digital intelligence for systems and machines (DISA) (pp. 143–150).

[28] Kumar, V., Boorla, K., Meena, Y., Ramakrishnan, G., Li, Y. F. (2018). Automating reading comprehension by generating question and answer pairs. In Phung, D., Tseng, V.S., Webb, G.I., Ho, B., Ganji, M., Rashidi, L. (Eds.) Advances in knowledge discovery and data mining (pp. 335–348).

[29] Tamura, Y., Takase, Y., Hayashi, Y., Nakano, Y. I. (2015). Generating quizzes for history learning based on Wikipedia articles. In Zaphiris, P., & Ioannou, A. (Eds.) Learning and collaboration technologies (pp. 337–346).