Automatic question generation using extended dependency parsing

Walelign Tewabe Sewunetie[1,2], Laszlo Kovacs[2]
[1]Department of Information Technology, School of Computing, Institute of Technology, Debre Markos University, Debre Markos, Ethiopia
[2]József Hatvany Doctoral School of Computer Science and Engineering, Faculty of Mechanical Engineering and Informatics, University of Miskolc, Miskolc, Hungary

Article Info

Article history:

Received Aug 25, 2023
Revised Oct 22, 2023
Accepted Dec 8, 2023

Keywords:

Automatic question generation
Dependency parsing
Evaluation metrics
Models of dependency parsing
Rule-based system

ABSTRACT

The importance of automatic question generation (AQG) systems in education is recognized for automating tasks and providing adaptive assessments. Recent research focuses on improving quality with advanced neural networks and machine learning techniques. However, selecting the appropriate target sentences and concepts remains challenging in AQG systems. To address this problem, the authors created a novel system that combined sentence structure analysis, dependency parsing approach, and named entity recognition techniques to select the relevant target words from the given sentence. The main goal of this paper is to develop an AQG system using syntactic and semantic sentence structure analysis. Evaluation using manual and automatic metrics shows good performance on simple and short sentences, with an overall score of 3.67 out of 5.0. As the field of AQG continues to evolve rapidly, future research should focus on developing more advanced models that can generate a wider range of questions, especially for complex sentence structures.

Corresponding Author:

Walelign Tewabe Sewunetie
József Hatvany Doctoral School of Computer Science and Engineering
Faculty of Mechanical Engineering and Informatics, University of Miskolc
Miskolc, Hungary
Email: walelightewabe@gmail.com

1. INTRODUCTION

Automatic question generation (AQG) has recently emerged as a valuable tool with the potential to transform educational assessments and automate tutoring. In a time when the demand for effective educational tools is on the rise, AQG offers a promising solution. AQG has the potential to reshape how educational assessments are conducted, offering the ability to generate questions from content, thereby reducing educators of time-consuming tasks [1]. However, the central challenge in developing effective AQG systems, as highlighted in previous research [2], revolves around the art of selecting target concepts and structuring questions around those concepts. These target concepts are the key ideas, topics, or pieces of information within a given text or context that AQG systems aim to create questions about. Target concepts are the crucial ideas, topics, or information within the text or context that AQG systems need to focus on. The choice of target concepts plays a pivotal role in generating meaningful questions. Question words, such as "who," "what," "where," "when," "why," and "how," determine the type and format of the questions. Selecting the appropriate question words is essential for coherence and context relevance in the generated questions.

The challenge lies in effectively identifying the right target concepts and question words for question generation. This is critical because the quality of the generated questions depends on these choices. Moreover, there is a need for a comprehensive approach that combines dependency parsing, adverb, and noun subtype selection, and rule-matching for scoring the generated questions.

To address this challenge, this paper presents a novel contribution by incorporating named entity recognition (NER), dependency parsing techniques, and adverb and noun subtype analysis, addressing a previously unexplored aspect. These subtypes include time, place, manner, degree, and frequency for adverbs, and human, animal, and thing for nouns. This novel approach identifies the relevant target concepts and question words, enhancing the quality of generated questions. Dependency parsing, a crucial task in natural language processing (NLP), analyzes the grammatical structure of sentences by establishing dependency relations between words [3]. Recent advancements in NLP have showcased the importance of dependency parsing in this context [4], [5]. This innovative combination allows for the generation of more contextually relevant and coherent questions.

As the field of AQG continues to evolve rapidly, future research should focus on developing more advanced models that can generate a wider range of questions, especially for complex sentence structures. The current system serves as a valuable foundation for further advancements in AQG, offering potential applications beyond educational settings. The implications of this research extend beyond the immediate scope, providing a stepping stone for future AQG developments.

The following sections will demonstrate how the integration of dependency parsing, NER, adverb, and noun subtype analysis improves the identification of target concepts and question words. Furthermore, we discuss how these improvements impact the quality of generated questions. This paper opens new avenues for AQG by addressing key limitations in existing systems. The integration of word2vec solutions for rule-matching calculation enhances the flexibility of this system, making it applicable across various domains.

2. DEPENDENCY PARSING

Dependency-based syntax, with functional relations, became more widely used in computational models compared to the phrase-structure-based constituency [6]. It identifies semantic connections between words in a sentence. It retrieves the sentence's syntactic structure from a linear sequence of word tokens by analyzing the relationships between words and determining each word's syntactic category. Recently, dependency-based syntactic parsing has gained popularity [7]. These parsers have been shown to work reliably for a broad range of languages [8]. The increased interest in dependency-based parsing has led to research into various parsing algorithms. The key difference between dependency and syntactic parsing is that dependency parsing builds a parse tree, while syntactic parsing constructs a syntax tree [9].

Dependency parsing is a crucial task in NLP, and recent years have seen significant advancements in this field [10]. According to Kübler *et al.* [9], dependency parsing models can be broadly classified into two major groups: grammar-based dependency parsing and data-driven dependency parsing. Grammar-based models are based on formal grammar and can be further divided into context-free dependency parsing and constraint-based dependency parsing. In contrast, data-driven approaches differ in the type of parsing model adopted, the algorithms used to learn the model data, and the algorithms used to parse new sentences with the model.

Data-driven dependency parsing models can be further categorized into transition-based and graph-based dependency parsing models [11]. Both transition-based and graph-based models are developed using supervised machine-learning techniques from linguistic data. Transition-based dependency parsing, also known as shift-reduce parsing, learns a model for scoring transitions from one parser state to the next, conditioned on the parse history. Parsing is then performed by greedily taking the highest-scoring transition out of every parser state until a complete dependency graph is derived. Figure 1 shows the example of transition-based dependency parsing, for a sentence “Budapest is the capital of Hungary”.

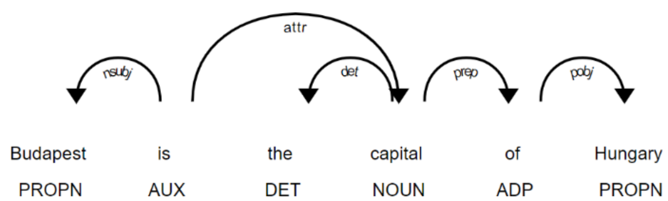

Figure 1. Dependency parsing example (Budapest is the capital city of Hungary)

The second important type of dependency parsing is the graph-based dependency parsing models, which were introduced by MacDonald and Nivre [12]. These models learn scoring functions in one shot and then perform an exhaustive search over the entire tree space for the highest-scoring tree. Currently, there are various dependency parsing tools available in Python that can be used to analyze sentence structure, some of the popular tools are spaCy, natural language toolkit (NLTK) with Stanford CoreNLP, and Stanza.

3. METHOD

Our methodology is grounded in a strategic fusion of dependency tree parsing and named entity recognition (NER) techniques. These choices are underpinned by their proven effectiveness and versatility in addressing the core challenges outlined in the introduction. Here, we provide the necessary details, algorithms, and techniques to allow readers to confirm and replicate our findings.

In this regard dependency tree parsing is a cornerstone of our approach and provides the means to analyze the grammatical structure of sentences by establishing dependency relations between words [13]. The choice of dependency parsing is justified by its inherent ability to handle various language constructs and ambiguous inputs effectively. Named entity recognition (NER) is another integral component of our methodology. NER automates the extraction of valuable information from unstructured natural language documents by categorizing named entities into predefined groups [14]. These groups include person names, organizations, locations, and more. Though conventional, we emphasize that these choices are essential to this method.

As reported by Mazidi and Tarau [15], dependency labels provide valuable information for extracting the meaning of the relationships between words. This technique constructs a tree structure that represents the syntactic dependency relationships between words, allowing us to identify the key semantic building blocks of the sentence. However, it was recognized that dependency parsing alone is insufficient for AQG, and additional tools like NER needed to be incorporated. GATE, OpenNLP, and spaCy are notable NER platforms [16]. For this study, spaCy NER was employed as a fast, statistical, and open-source named entity visualizer. The system assigns labels to groups of contiguous tokens, which encompass named or numerical entities, including person, organization, language, and event, among others [17].

Our proposed system is illustrated in Figure 2 and is categorized into distinct modules: pre-processing: the initial module, involving the removal of stop words and tokenization of the remaining words from the input sentence. NER, POS, and dependency parsing: the subsequent modules process the tokenized data, identifying named entities, extracting parts of speech (POS) tags, and performing dependency parsing. These elements form the foundation for subsequent stages. The output of this module serves as input for the NER, POS, and dependency parsing modules. The NER module identifies named entities within the input, while the POS module extracts the noun components of the sentence, which are also essential for the ruleset mapping and question generation stages.

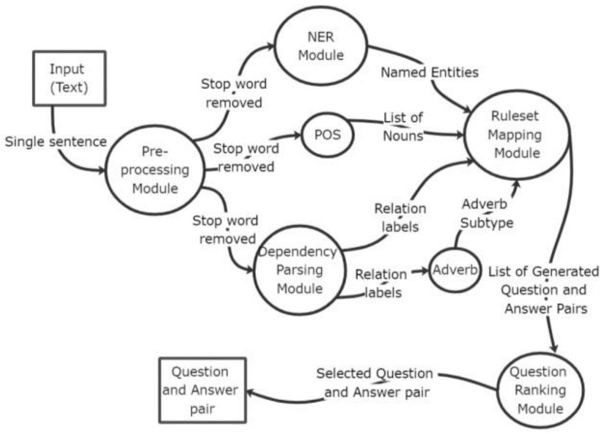

Figure 2. Proposed system block diagram

The Ruleset adopted from previous work [18], is extended to include named entities, POS tags, and dependency parsing [19]. This enhancement acknowledges the importance of these elements for generating high-quality questions. However, the main limitation of the ruleset was its lack of categorization for adverbs and noun types.

To address this limitation, we present Algorithms 1 and 2, which depict the essential steps in our methodology: Algorithm 1: ruleset mapping for question generation: this algorithm maps rules to dependency tag lists and selects the best matching rule. It is an essential component of our innovative approach. Algorithm 2: question word selection for question generation: this algorithm determines the appropriate question word (Wh_QTypeWord) based on inputs, including NER, adverb subtype, noun subtype, and dependency tags. This step contributes significantly to question generation.

Algorithm 1. Ruleset mapping for question generation algorithm
```
Function RulesetMapping
Input:= Ruleset,
List_of_Sent_DependencyTag
Output:=Question, Answer
Begin
QuestionList←empty
For R←Rule to Ruleset do
Sim←similarity (Rule, DependenyTagList)
If sim is in Bestsimilarityscore then
WinnerRule←Rule
Bestsimilarityscore←sim
End if
End for
QuestionList←apply(WhQTypeWord, WinnerRule, DependencyTagList)
Return QuestionList
End
Function
```

In response to the limitations of conventional rule-based systems, our methodology innovatively integrates word2vec [20], a powerful word embedding technique. This integration augments the flexibility and effectiveness of our system, making it applicable across diverse domains. The authors noted that the state-of-the-art best match analysis calculation is commonly used to perform rule-set matching. Nevertheless, this mechanism for selecting the best match is rigid, and there are numerous scenarios in which sentences may express the same meaning but are written differently.

Algorithm 2. Question word selection for question generation algorithm
```
Function QuestionGeneration
Input:=Ruleset,
List_of_sent_NER,
List_of_sent_AdverbSubType,
List_of_sent_NounSubType,
List_of_sent_DependencyTag
Output:=Wh_QType
Begin
Set DependencyTagList←ListofSentenceDependencyTags
Set QuestionList← Empty
BestSimilarityscore=empty
Wh_QTypeWord=empty
BestScore=empty
if List_of_Sent_NER NOT empty then
if List_of_Sent_NER==”PERSON” then
Wh_QTypeWord=”Who”
Else if List_of_Sent_NER==”LOC” then
Wh_QTypeWord=”What”
Else if List_of_Sent_NER==”DATE” then
Wh_QTypeWord=”When”
…
End if
Else if List_of_Sent_AdverbSubType NOT empty then
If List_of_Sent_AdverbSubType==”PLACE” then
Wh_QTypeWord=”Where”
Else if List_of_Sent_AdverbSubType==”TIME” then
Wh_QTypeWord=”When”
Else if List_of_Sent_AdverbSubType==”MANER” then
Wh_QTypeWord=”How”
Else if List_of_Sent_AdverbSubType == ”FREQUENCY” then
Wh_QTypeWord=”How Often”
…
Else
Endif
```

```
Else
If List_of_Sent_NounSubType==”PERSON” then
Wh_QTypeWord=”Who”
Else if List_of_Sent_NounSubType==”ANIMAL” then
Wh_QTypeWord=”What”
Else if List_of_Sent_NounSubType==”OBJECT” then
Wh_QTypeWord=”Which”
…
Else
End if
End if
return Wh_QTypeWord
RulesetMapping()
End Function
```

A distinctive feature of our approach is the inclusion of adverb subtypes (time, place, manner, degree, and frequency) [21] and noun subtypes (human, animal, and thing) for question generation. These subtypes play a pivotal role in crafting high-quality questions. We provide comprehensive tables (Tables 1 and 2) that detail the combinations of these subtypes with their corresponding question words.

Table 1. Noun SubTypes and corresponding question words

Noun subtype	Question word
Human	Who
Animal	What
Thing	What

Table 2. Adverb SubTypes and corresponding question words

Adverb subtypes	Question word
Time	When
Place	Where
Manner	How
Degree	How
Frequency	How often

4. RESULTS AND DISCUSSION

In recent years, numerous models, and evaluation techniques for AQG have been presented in the literature. Various metrics, including bilingual evaluation understudy (BLEU), ROUGE, and metric for evaluation of translation with explicit ordering (METEOR), have been utilized to evaluate the effectiveness of AQG systems by measuring the similarity of the generated questions to the reference questions [22], [23]. However, the evaluation process also gave significant importance to the answerability and naturalness of the questions, as they are essential factors in determining the quality of generated questions [24].

A human evaluation was conducted to assess the answerability of the generated questions, with the significance of question types (Wh-types), named entities, and content words (often relations) being determined in various AQG tasks. Furthermore, the grammar structure and naturalness of the generated questions were considered fundamental parameters in human evaluation:

David ate an apple on Monday
    Who ate apple on Monday?
Menna typed on the computer keyboard
    Who typed on the computer keyboard?
Budapest is the capital city of Hungary
    What is the capital city of Hungary?
    Which country Budapest is?
The New Year of Ethiopia is on September
    When the New Year is?
    What is on September?
Ethiopian New year is on September
    When Ethiopian New Year is?
    What is on September?

The evaluation process involved a Google Form questionnaire that allowed participants to rate the generated questions on a scale of 1 to 5, where 1 denoted poor and 5 indicated excellently. The overall outcome of the human evaluation was encouraging, with a score of 3.67.

The researchers compare system-generated questions with human-generated questions and use automatic evaluation techniques. The results of the evaluation, as it is presented in Table 3, suggest that the system performs well, especially in short sentences. The average BLEU-N score of 0.718 indicates that the system-generated questions have a reasonable level of similarity to the human-generated questions. However, it is important to keep in mind that limitations exist with BLEU-N scores, and they may not necessarily reflect the quality of the questions in terms of their informativeness, relevance, and coherence.

Table 3. BLEU–N and ROUGE-N metrics of automatic evaluation result

Metrics	Type	Score
BLEU	1-gram	0.862654
	2-gram	0.785234
	3-gram	0.773411
	4-gram	0.751432
Rouge-1	F1 score	0.619192
	Precision	0.59619
	Recall	0.65
Rouge-2	F1 score	0.533333
	Precision	0.52
	Recall	0.55
Rouge-L	F1 score	0.619192
	Precision	0.59619
	Recall	0.65

The ROUGE score, on the other hand, is based on a different metric that measures the overlap between system-generated and human-generated questions in terms of n-gram sequences. The fact that ROUGE had the highest F1-score suggests that the system-generated questions had a high level of overlap with the human-generated questions in terms of n-gram sequences, although it doesn't consider different words with the same meaning. It was revealed through the experimental analysis that the combination of dependency parsing with NER is effective in identifying the subject, verb, object, and adverb parts of a sentence [25], which are essential for question generation. The effectiveness of identifying the subject, verb, object, and adverb parts of a sentence, which are essential for question generation [26], is revealed through experimental analysis of the combination of dependency parsing with NER. For instance, consider the following two sentences, which have the same meaning and can generate the same question. The subject, verb, object, and adverb parts of a sentence are extracted using dependency parsing in the system.

5. DISCUSSION

AQG has seen significant advancements in recent years, with the development of various models that use deep learning techniques to generate questions from different types of textual data. For instance, the recent works of Zhao *et al.* [27] have proposed neural network-based models that utilize contextual embedding and attention mechanisms for question generation. Furthermore, a crucial NLP objective is to extract significant sentences from a given text, and another objective is to generate extractions based on the original text. In this context, rule-based systems play a vital role in extracting pertinent words for generating uncomplicated and domain-specific questions.

In this paper, a rule-based AQG system was proposed that employs dependency parsing and considers various types of wh-question words. The system uses a combination of NER, POS, dependency tags, and adverb subtypes for rule-set mapping and question generation. While the proposed system has demonstrated good performance for simple sentence structures, future research could explore the integration of neural network-based models to improve the system's ability to generate complex questions. Moreover, future work could also focus on enhancing the system to include paragraph-based question generation. Overall, the proposed rule-based system provides a foundation for developing more sophisticated AQG systems that can generate questions from a wide range of textual data.

6. CONCLUSION

In conclusion, this paper presented a rule-based AQG system that utilized dependency parsing and a comprehensive analysis of English sentence structure. The system was evaluated using both automatic and human evaluation techniques, and the results showed that the quality of the generated questions was highly dependent on the complexity of the sentence, with better quality and more natural questions generated for sentences with simple structures. Recent advances in AQG have led to the introduction of new models that utilize machine learning techniques, including neural networks, to generate questions from the text. These models can generate questions from both single sentences and paragraphs and have the potential to generate more complex and diverse questions. Furthermore, machine learning techniques, including neural networks, have been applied to question-generation models for various domains, including medical and scientific question generation. In conclusion, the field of AQG is rapidly evolving, and future work will likely focus on developing more advanced models that can generate more diverse and complex questions. The current rule-based system presented in this paper serves as a baseline for future research in the field.

ACKNOWLEDGEMENTS

The preferred spelling of the word "acknowledgment" in America is without an "e" after the "g". Avoid the stilted express the research reported here was carried out as part of the EFOP-3.6.1-16-2016-00011 "Younger and Renewing University-Innovative Knowledge City-Institutional Development of the University of Miskolc aiming at intelligent specialization" project implemented in the framework of the Széchenyi 2020 program. The realization of this project is supported by the European Union and co-financed by the European social fund.

REFERENCES

[1] O. Keklik, T. Tuglular, and S. Tekir, "Rule-based automatic question generation using semantic role labeling," *IEICE Transactions on Information and Systems*, vol. E102.D, no. 7, pp. 1362–1373, Jul. 2019, doi: 10.1587/transinf.2018EDP7199.

[2] L. Kovács and W. T. Sewunetie, "Challenges in target selection for automatic question generation," *Multidiszciplináris Tudományok*, vol. 9, no. 4, pp. 500–507, 2019, doi: 10.35925/j.multi.2019.4.53.

[3] S. Oepen *et al.*, "SemEval 2015 task 18: broad-coverage semantic dependency parsing," in *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, Stroudsburg, PA, USA: Association for Computational Linguistics, 2015, pp. 915–926. doi: 10.18653/v1/S15-2153.

[4] Z. Li, J. Cai, S. He, and H. Zhao, "Seq2seq dependency parsing," in *COLING 2018 - 27th International Conference on Computational Linguistics, Proceedings*, 2018, pp. 3203–3214.

[5] S. Tratz and E. Hovy, "A fast, accurate, non-projective, semantically-enriched parser," in *EMNLP 2011-Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, 2011, pp. 1257–1268.

[6] A. Novák and B. Novák, "Cross-lingual transfer of knowledge in distributional language models: experiments in Hungarian," *Acta Linguistica Academica*, vol. 69, no. 4, pp. 405–449, Dec. 2022, doi: 10.1556/2062.2022.00580.

[7] J. Nivre, A. Basirat, L. Dürlich, and A. Moss, "Nucleus composition in transition-based dependency parsing," *Computational Linguistics*, vol. 48, no. 4, pp. 849–886, Dec. 2022, doi: 10.1162/coli_a_00450.

[8] C. Alzetta, F. Dell'Orletta, S. Montemagni, and G. Venturi, "Linguistically-driven selection of difficult-to-parse dependency structures," *Italian Journal of Computational Linguistics*, vol. 6, no. 2, pp. 37–60, Dec. 2020, doi: 10.4000/ijcol.719.

[9] S. Kübler, R. McDonald, and J. Nivre, "Dependency parsing," in *Dependency Parsing. Synthesis Lectures on Human Language Technologies*, Cham: Springer, 2009, pp. 11–20. doi: 10.1007/978-3-031-02131-2_2.

[10] T. Osborne, "NPs, not DPs: the NP vs. DP debate in the context of dependency grammar," *Acta Linguistica Academica*, Mar. 2021, doi: 10.1556/2062.2021.00001.

[11] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "BLEU: a method for automatic evaluation of machine translation," in *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics-ACL '02*, Morristown, NJ, USA: Association for Computational Linguistics, 2002, pp. 311–318, doi: 10.3115/1073083.1073135.

[12] R. McDonald and J. Nivre, "Analyzing and integrating dependency parsers," *Computational Linguistics*, vol. 37, no. 1, pp. 197–230, Mar. 2011, doi: 10.1162/coli_a_00039.

[13] A. A. Yusuf, N. A. Nwojo, and M. M. Boukar, "Basic dependency parsing in natural language inference," in *2017 13th International Conference on Electronics, Computer and Computation (ICECCO)*, IEEE, Nov. 2017, pp. 1–4, doi: 10.1109/ICECCO.2017.8333337.

[14] I. El Bazi and N. Laachfoubi, "Arabic named entity recognition using deep learning approach," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 9, no. 3, p. 2025, Jun. 2019, doi: 10.11591/ijece.v9i3.pp2025-2032.

[15] K. Mazidi and P. Tarau, "Infusing NLU into automatic question generation," in *Proceedings of the 9th International Natural Language Generation conference*, Stroudsburg, PA, USA: Association for Computational Linguistics, 2016, pp. 51–60, doi: 10.18653/v1/W16-6609.

[16] N. Tsinganos and I. Mavridis, "Building and evaluating an annotated corpus for automated recognition of chat-based social engineering attacks," *Applied Sciences*, vol. 11, no. 22, p. 10871, Nov. 2021, doi: 10.3390/app112210871.

[17] R. Srihari and W. Li, "Information extraction supported question answering," in *Cymfony Net Inc Williamsville Ny*, 1999, pp. 1–6.

[18] Ó. Egyetem and S. E. M. Tudományegyetem, *IEEE Joint 22nd International Symposium on Computational Intelligence and Informatics and 8th International Conference on Recent Achievements in Mechatronics, Automation, Computer Science and Robotics proceedings : 2022, November 21-22, Budapest, Târgu Mureş*, 2022.

[19] S. M. A. El-Morsy, M. Hussein, and H. M. Mousa, "Arabic open information extraction system using dependency parsing," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 12, no. 1, p. 541, Feb. 2022, doi: 10.11591/ijece.v12i1.pp541-551.

[20] M. A. Fauzi, "Word2Vec model for sentiment analysis of product reviews in Indonesian language," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 9, no. 1, p. 525, Feb. 2019, doi: 10.11591/ijece.v9i1.pp525-530.

[21] R. W. Fasold and J. Connor-Linton, *An Introduction to Language and Linguistics*, no. 37. Cambridge University Press, 2006.

[22] A. Gašpar, A. Grubišić, and I. Šarić-Grgić, "Evaluation of a rule-based approach to automatic factual question generation using syntactic and semantic analysis," *Language Resources and Evaluation*, vol. 57, no. 4, pp. 1431–1461, Dec. 2023, doi: 10.1007/s10579-023-09672-1.
[23] N. Mulla and P. Gharpure, "Automatic question generation: a review of methodologies, datasets, evaluation metrics, and applications," *Progress in Artificial Intelligence*, vol. 12, no. 1, pp. 1–32, Mar. 2023, doi: 10.1007/s13748-023-00295-9.
[24] B. Bohnet, "Comparing advanced graph-based and transition-based dependency parsers," in *Proceedings of Depling*, 2011, pp. 282–289.
[25] N. Afzal and R. Mitkov, "Automatic generation of multiple choice questions using dependency-based semantic relations," *Soft Computing*, vol. 18, no. 7, pp. 1269–1281, Jul. 2014, doi: 10.1007/s00500-013-1141-4.
[26] F. Maheen *et al.*, "Automatic computer science domain multiple-choice questions generation based on informative sentences," *PeerJ Computer Science*, vol. 8, p. 1010, Aug. 2022, doi: 10.7717/peerj-cs.1010.
[27] Y. Zhao, X. Ni, Y. Ding, and Q. Ke, "Paragraph-level neural question generation with maxout pointer and gated self-attention networks," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, Stroudsburg, PA, USA: Association for Computational Linguistics, 2018, pp. 3901–3910. doi: 10.18653/v1/D18-1424.

BIOGRAPHIES OF AUTHORS

Walelign Tewabe Sewunetie received the B.Sc. degree in computer science from the Mekelle University, Ethiopia, and the M.Sc. degree in Computer Science from the Arba Minch University, Ethiopia. He was the deputy registrar director of Debre Markos Institute of Technology, Debre Markos University, Ethiopia. He was also acting as the managing director of Debre Markos Institute of Technology at Debre Markos University, Ethiopia from 2018 to August 2019. He is currently a Ph.D. candidate at the University of Miskolc, Hungary. His research interests include soft computing, machine learning, and intelligent systems. He can be contacted at email: walelingtewabe@gmail.com.

László Kovács is a full professor at the University of Miskolc, Hungary, Institute of Information Sciences. He is the Department Head of Software Engineering. His main research area includes database and knowledge base modeling, concept lattice structures, discrete optimizations, and machine learning algorithms in NLP. He has authored or coauthored more than 194 publications: conference 109 proceedings and 85 journals. He has supervised more than 10 Ph.D. students. He can be contacted at email: kovacs@iit.uni-miskolc.hu.