Aganittyam: Learning Tamil Grammar through Knowledge Graph based Templatized Question Answering

Mithilesh K, Amarjit Madhumalararungeethayan, Dharanish Rahul S, Abhijith Balan, C Oswald, Hrishikesh Terdalkar

Proceedings of the 38th Pacific Asia Conference on
Language, Information and Computation (PACLIC 38)

Nathaniel Oco, Shirley N. Dita, Ariane Macalinga Borlongan, Jong-Bok Kim (eds.)

2024

Aganittyam: Learning Tamil Grammar through Knowledge Graph based Templatized Question Answering

Mithilesh K[1], Amarjit Madhumalararungeethayan[1], Dharanish Rahul S[1],
Abhijith Balan[1], C Oswald[1], and Hrishikesh Terdalkar[2]

[1]Dept. of Computer Science and Engineering, National Institute of Technology, Tiruchirappalli, India.
[2]LIRIS Research Lab, University of Lyon 1, France
{106120069, 106120011, 106120031, 406123001, oswald}@nitt.edu
hrishikesh.terdalkar@univ-lyon1.fr

Abstract

In this work, we present a novel Grammar Question-Answering System, Aganittyam, along with its associated corpus focused on the Dravidian language Tamil. As one of the oldest surviving languages with a documented history exceeding 2,000 years, Tamil is recognized as a classical language and holds official status in three countries, including India, while being spoken by various diasporic communities worldwide. Learning Tamil grammar poses challenges due to its agglutination and complex morphology. Despite the active research in automatic processing of Tamil texts, there are currently no automated tools available to assist learners. To address this gap, we created a comprehensive corpus of Tamil grammar designed to facilitate learning. We developed an ontology comprising 7 relationship types, manually annotating the corpus to identify entities and relationships. The resultant triplets (subject–predicate–object) were organized into a knowledge graph (KG) consisting of $63,587$ entities. Our framework, *Aganittyam*, enables template-based question-answering, providing a structured approach to learning. We conducted a bifold evaluation—incorporating both query metrics and human-centric assessments—demonstrating that our QA system is robust, reliable, and engaging for answering various objective questions. The system is available at https://aganittyam-web.onrender.com/home.

1 Introduction

The concept of knowledge graph (KG) was initially proposed by Google. A knowledge graph is a large-scale knowledge base composed of a large number of entities and relationships between them (Fensel et al., 2020b; Chen et al., 2020b; Kejriwal et al., 2021). A structured representation of facts, consisting of entities, relationships, and semantic descriptions is maintained. A KG primarily consists of two components (node and edge) where a node represents an entity and edge represents relationship between nodes. A sample KG and its illustration is given in Figure 1 (KG). It illustrates the fact that *Bob* is *interested in Monalisa* and *Mona Lisa* was *created by Leonardo Da Vinci*. There are many applications of knowledge graphs such as Question Answering System, Recommender System and Information Retrieval etc. A question answering (QA) system's main objective is to use facts in the knowledge graph (KG) to answer natural language questions. Most of the extant QA systems for Indian languages focus on Hindi. There is a paucity of research work in QA systems for Dravidian languages, primarily attributing to the limited number of dataset available in these languages.

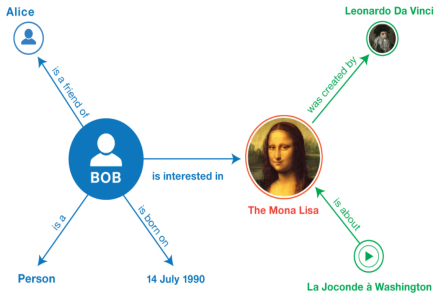


Figure 1: A sample Knowledge Graph

1.1 Motivation towards Tamil Language and Grammar

Tamil, with its rich literary heritage spanning over two millennia, features a unique and intricate grammatical system (Sarveswaran, 2024; Asher, 1985). For many learners, especially non-native speakers, mastering this system can be daunting due to its complex phoneme structure, distinctive script with intricate characters, phonetic nuances, and diverse regional variations. The grammar reflects the language's long history, encompassing key features such as word formation (morphology), sentence structure (syntax), verb conjugation, nouns, pronouns, postpositions, and verb-noun constructions (Steever, 2018; Sarveswaran, 2024; Asher, 1985).

The challenges in learning Tamil grammar include agglutination, complex morphology, an extensive case system, and rich vocabulary, along with pronunciation and phonology. Moreover, existing teaching methods often present grammatical concepts in a disjointed manner, hindering comprehension and appreciation of the language's depth. Students, in particular, tend to find these concepts more challenging than adults.

Currently, some NLP tools available for grammar learning include POS taggers, chunkers, dependency parsers, morphological analyzers, and morphological generators (Singh and Shah, 2022; Rajendran et al., 2022; Dhanalakshmi et al., 2010). However, to the best of our knowledge, no tool exists that facilitates an interactive approach to learning Tamil grammar while effectively assessing understanding of the basic concepts.

1.2 Knowledge Graphs for Tamil Grammar Question Answering

In recent years, deep learning models have been developed for extractive question answering systems in the Tamil language (Krishnan et al., 2023a). For instance, (Murugathas and Thayasivam, 2022) introduced a question answering system comprising multiple modules, specifically trained on a manually tagged dataset focused on the historical domain in Tamil. This innovative approach highlights the potential of leveraging tailored datasets and modular designs to enhance the accuracy and relevance of responses in Tamil question answering.

While deep learning models have their merits, knowledge graphs offer distinct advantages, including transparency, explicit domain representation, consistency with expert knowledge, semantic understanding, logical reasoning, scalability, and support for complex queries (Futia and Vetrò, 2020; Turing Institute). We aim to bridge this gap by developing a knowledge graph dataset for templated question answering that aids grammar learning, leveraging the strengths of these powerful tools to refine ontologies. An ontology is a formal, structured representation of knowledge within a specific domain (Guarino et al., 2009), defining the concepts, entities, and relationships, along with their interactions. Ontologies facilitate better understanding, sharing, and reuse of information across systems. Knowledge graphs provide a structured, visual, and scalable way to represent and explore complex relationships crucial for accurate ontology development. However, literature on knowledge graphs specific to the Tamil language is scarce, and to the best of our knowledge, there has been no work on constructing a Tamil grammar-based knowledge graph.

1.3 Salient Features

This research aims to develop a novel Tamil grammar question-answering system by creating a comprehensive Tamil grammar corpus, performing human annotation, and constructing a Tamil Grammar Knowledge Graph. This Knowledge Graph will facilitate templatized question answering, providing a dynamic and interactive learning experience. Our system not only helps learners grasp Tamil grammar but also assesses their skills in a motivating way. As new grammatical concepts emerge, they can be easily incorporated, ensuring the resource remains relevant and up to date.

1.4 Contributions

Our main contributions are as follows:

- Created a *Tamil Grammar Corpus* from web sources, featuring $63,587$ entities and relations between them adhering to 7 major relation types.
- Developed a straightforward method for constructing a richly *human-annotated Tamil grammar knowledge graph* and its corresponding ontology.
- Introduced *"Aganittyam", a templatized question-answering tool* that generates grammar questions, including complex

queries—marking the first tool of its kind for Tamil.

- Conducted rigorous evaluations of the QA tool using both *Query Evaluation metrics* and *Human-Computer Interaction metrics*.

2 Architecture

In this section, a detailed description about the KG construction and templatized question-answering technique for Tamil grammar are provided. Figure 2 illustrates the complete architecture of the proposed *Aganittyam*. Few snapshots of the same can be seen in Figures 6 and 7 in in appendix C. We use an annotation tool, *Sangrahaka* (Terdalkar and Bhattacharya, 2021), for the construction of knowledge graph.

2.1 Tamil Grammar Corpus Construction

To the best of our knowledge, there does not exit a Tamil Grammar dataset targeted for an NLP task. The source of our dataset construction includes (Tamil Wikipedia) (Tamil Wikinaotinary) and (Byjus Page for TN Books).

Tamil ilakkaṇam (தமிழ் இலக்கணம்) is the name of the corpus uploaded in *Sangrahaka* as shown in Figure 8 in appendix C. In *Sangrahaka*, the administrator has the privilege to insert, delete and update the corpus, and can provide the details of corpus in the UI. The corpus exhibits numerous relation types across sentences. The following are some of notable relation types with examples.

i. பெயர்ச்சொல்(peyarchchol) - Noun
   - பொதுவான பெயர்சொற்கள் (Potuvān̲a peyarccor̲kaḷ) - Common Nouns (வங்கி(bank), பாடசாலை(school))
   - சரியான பெயர்ச்சொற்கள் (Cariyān̲a peyarccor̲kaḷ) - Proper Nouns (இலண்டன்(London), மதுரை(Madurai))
   - திடப் பெயர்சொற்கள் (Tiṭap peyarcor̲kaḷ) - Concrete Nouns (மரம்(tree), பந்து(ball))
   - நுண் பெயர்ச்சொற்கள் (Nuṇ peyarccor̲kaḷ) - Abstract Nouns (திறமை(skill), கருத்து(opinion))

ii. எதிர்ச்சொல்(ethirchchol) - Antonyms

iii. இணைப்பொருட்ச்சொற்கள் (iṇaipporuṭccor̲kaḷ) - Synonyms

iv. ஒருமை பன்மை(orumai pan̲mai) - Singular Plural

v. சேர்த்து எழுதுக(cērttu el̲utuka) - Words Join

vi. காலங்கள்(kālaṅgaḷ) - Tenses
   - We have three tenses: Past, Present and Future Tense.

vii. பிரித்தல் (pirithal) - Words Split

To the best of our knowledge, we have gathered all publicly available resources (sentences) for each relation type. Our Tamil Corpus consists of 7 relation types and $63,587$ entities (words).

2.2 Ontology Construction and Annotation

Ontology refers to structured representation of knowledge about a domain which forms the skeleton of a knowledge graph (Estival et al., 2004). Ontology construction, for Tamil grammar KG is managed by *Sangrahaka* as illustrated in Figure 9 in Appendix C. Figure 3 showcases the working of ontology where nodes are labeled as *Words* and edges represent grammatical relations. For an example, entities முட்டாள்(muttaal(stupid)) and புத்திசாலி(buddhisali(intelligent)) represent the relation type இணைப்பொருட்ச்சொற்கள்(inaippoutchorkal(synonyms)) and entities தைரியமான(tairiyamaana(courageous)) and துணிச்சலான(thunichalaana(brave)) represents the relation type எதிர்ச்சொல்(ethirchchol(antonym)).

Once the preprocessing steps such as tokenization and segmentation are performed, the annotation process of individual words or phrases are assigned with their appropriate relation type. During annotation phase, edges are assigned as relation types and nodes as entities as shown in Figure 10, 11 and 12 in appendix C.

2.2.1 Question Templates and Triplets

Triplets represent real-world facts and semantic relations in a knowledge base. In a knowl-

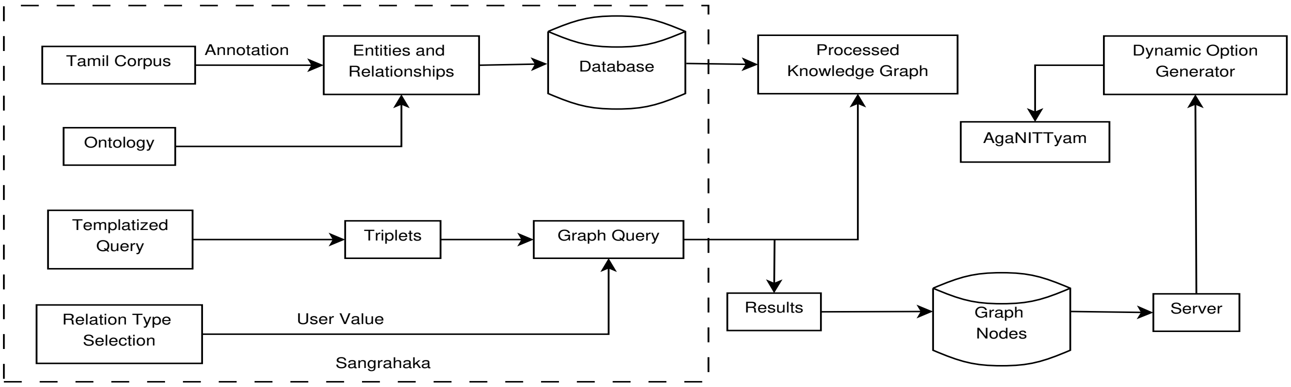


Figure 2: Architecture of our proposed Tamil Grammar Learning through Knowledge Graph based Templatized Question-Answering

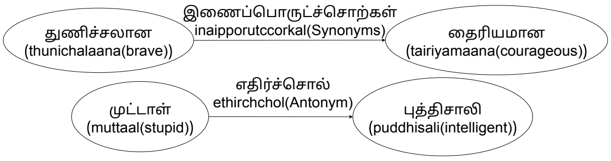


Figure 3: Example of Relations in KG

edge graph, a triplet is an edge between two nodes, where the edge represents a relation and the nodes represent entities. In the case of Tamil grammar KG, entities represents the individual words and edges represents grammatical relations. Once the triplets are identified, based on triplets, the templates are generated for constructing Knowledge Graph. Some of the relation types, their triplets and the question templates are given below.

Relation Type: ஒருமை பன்மை (orumai pan̲mai) (Singular Plural)
Triplet: (பந்து (Ball), is-Plural, x)
Template: (பந்து(Ball) + is-Plural = x)

Relation Type: சேர்த்து எழுதுக (cērttu-el̲utuka) (Joined Words)
Triplet: (கரும்பு(Karumpu), +, சாறு(Cār̲u))
Template: (x + y = z)

Relation Type: Complex Query
Triplet: ((கை (Arm), is-Noun, (Yes or No)), is-Plural, x)
Template: ((கை (Arm), is-Noun) ? Yes:No) is-Plural = x)

2.2.2 Complex Queries

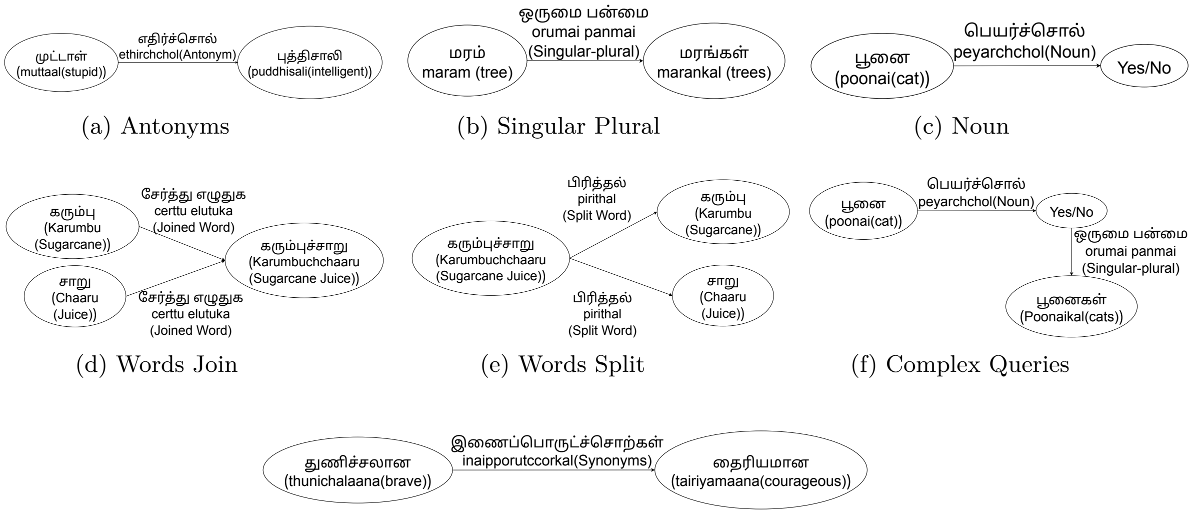
Complex queries helps in understanding and reasoning about the connections between different pieces of data. For an example in case of Tamil grammar corpus creation, we designed complex queries which tells whether given word is noun or not. In case if it is noun, then it outputs the plural form of the particular word. An example is given below.

Question: அணி என்பது பெயர்ச்சொல்லா? அப்படியானால், அதன் பன்மை வடிவம் என்ன? (An̲i en̲patu peyarccollā? Appaṭiyān̲al, atan̲ pan̲mai vaṭivam en̲n̲a?) (Is ani a noun? If so, what is the plural form of it?)
Template: ((அணி (Ani) (Team), is-Noun) ? Yes:No) is-Plural = x)

Question: பெண்ணின் ஆண்பால் பன்மை என்ன? (What is the plural of masculine of girl?)
Template: ((பெண்ணின்(Peṇṇin̲)(Girl),-Gender) ? Male:Female) is-Plural = x)

2.3 KG Construction

The knowledge graph is constructed by manual annotation with the help of two annotators. Both the annotators are native Tamil speakers with adequate knowledge of Tamil grammar. Words are annotated as entities and grammatical relationships as edges. The KG is stored in a graph database, with annotations converted into a machine-readable format using a Python script. Figure 4 provides examples of different relation types and their corresponding knowledge graph. This knowledge graph can be used for analysis, exploration, and discovery of information within the corpus.



(a) Antonyms

(b) Singular Plural

(c) Noun

(d) Words Join

(e) Words Split

(f) Complex Queries

(g) Synonyms

Figure 4: A sample Knowledge Graph constructed for relation types in Tamil Grammar

2.4 *Aganittyam* - The Question Answering Portal for Tamil Grammar

We named the QA portal *Aganittyam*, from the word *Agattiyam* (the earliest book on Tamil Grammar)(Agattiyam Wikipedia), which is a dynamic platform to learn Tamil grammar using templatized question-answering.

2.4.1 Features of Aganittyam

It allows diverse question categories on all the 7 types of relations, dual language support (English to Tamil and vice-versa) for non-native Tamil speakers and interactive exercises ensuring that options are generated as per their category choice. By harnessing the power of the Tamil KG in the QA Tool, the learner is not only equipped with practical language skills but also gains a deeper understanding of the intricate connections within Tamil grammar.

2.4.2 Architecture of *Aganittyam*

Aganittyam relies on a robust framework to deliver effective, interactive, and personalized learning solutions. Dynamic option generation, Tamil Knowledge Graph, frontend interface and backend infrastructure are the key components of our Aganittyam.

3 Results and Discussions

To the best of our knowledge, there does not exist a Tamil Grammar QA Systen to compare with our work. Simulation is performed on an AMD Ryzen 7 CPU with 16 GB Main Memory and 512 GB Hard Disk on Windows 10 Platform. Using Python and Anaconda tool, the proposed techique was implemented and Neo4j Database was used to store the KG.

Our experimentation on KG based Tamil Grammar QA system is bi-fold. On one side, to test the performance of the QA system, we thoroughly experimented with various templatized queries. On the other side, a detailed User Satisfaction studies were carried out with four different metrics.

3.1 KG based Experimental Results

The following are the metrics used to evaluate our KG based tamil Grammar QA tool. Table 1 illustrates the experimental results of the KG based performance metrics for each of the relation types in the Knowledge Graph.

3.1.1 Accuracy

Accuracy of the result of a Query is defined as follows:

$$\text{Accuracy} = \frac{\text{Number of Correct Query Retrievals}}{\text{Total Query Retrievals}}$$

Accuracy measures the precision and correctness of the information fetched using cypher-queries from the Neo4j database, ensuring that the likelihood of correct responses is maximized. A higher accuracy implies an efficient system. In the evaluation process, the accuracy of data retrieval is assessed across various categories pertinent to the system's functionality. Each category represents a distinct aspect or set of queries within the system. A total of 1000 query retrieval iterations were performed for each Relation type, for which each results were checked for correspondence with the ground truth. The experimental results for accuracy of the query results are presented in Table 1. The total average data accuracy of the portal was found out to be 94%. The reduction in performance is due to human error during annotation and dataset creation, Tamil character encoding limitations and invalid JSON parses.

3.1.2 Knowledge Graph Utilization Ratio (KGUR)

The Knowledge Graph Utilization Ratio (KGUR) is defined as follows:

$$\text{KGUR} = \frac{\text{Number of subgraphs in Knowledge Graph}}{\text{Total Number of nodes in Knowledge Graph}}$$

This quantifies the extent to which the Knowledge Graph is utilized in generating responses, reflecting the system's reliability on structured knowledge for answering queries. A higher KG Utilization Ratio signifies a larger number of relationships existing between nodes. This metric evaluates how effectively the system leverages structured knowledge, reflecting its reliance on interconnected data to generate responses. The best KG Utilization Ratio achieved was around 0.433 where the number of subgraphs in Knowledge Graph is 580 and the total Number of Nodes is 63,587.

3.1.3 Average Query Response

The Average Query Response is defined as follows:

$$\text{Average Query Response} = \frac{\sum_{i=1}^{n}(Q_i)}{\text{Total Query Retrievals } (n)}$$

where Q_i represents the average response time in the i^{th} retrieval. It evaluates the efficiency of the system in responding to user queries within a specified timeframe, indicating its responsiveness and ability to handle user interactions promptly. A lower Average Query Response represents that the cypher-queries are optimized, resulting in faster page loads. Average Query Response measures the system's responsiveness by evaluating the speed at which it handles user queries. A lower response rate indicates faster query processing, contributing to a smoother user experience and quicker access to information. A total of 1000 query retrievals were made for each category to arrive at this conclusion. It is observed that the average query response of the portal was found out to be around 2.48 seconds.

3.1.4 Degree of Randomness

The Degree of Randomness (DR) is defined as follows:

$$\text{DR} = \frac{\text{Count of Distinct Nodes Retrieved}}{\text{Total Query Retrievals (n)}}$$

Degree of Randomness assesses the level of unpredictability or variability in the system's responses, providing insights into the system's ability to generate diverse and contextually relevant answers. A higher Degree of Randomness indicates a much more diverse dataset giving the user a better experience. One primary observation made here is the fact that the degree of randomness of a category is directly proportional to the size of the category's dataset. Similar to other performance metrics, the script for degree of randomness was made to run for a total of 1000 iterations (categorically).

3.2 User Satisfaction Metrics

The following metrics are entirely based on results derived from a survey by using *Aganittyam* tool conducted on approximately 500 school and college students. It was primarily directed towards school students who currently learn Tamil Grammar, and further expanded to college students as well. In this section, a detailed description about a set of metrics that provide valuable insights of user's perceptions, experiences and engagement is given. The following metrics are used to measure user satisfaction and scaled over a scale of 1 to 5 where 5 represents high positive value.

3.2.1 Customer Effort Score (CES)

CES measures the level of effort required by customers to interact with the system. Figure 5a shows the Audience Percentage Split in

Relation Types-*Aganittyam*	Accuracy of Query Retrieval	Query Response Time (in seconds)	Degree of Randomness
Noun Classification	0.92	2.41	0.90
Synonyms	0.99	2.51	0.67
Antonyms	1.0	2.49	0.83
Singular/Plural	0.99	2.47	0.87
Word Split	0.99	2.52	0.65
Tenses	0.99	2.54	0.59
Complex Queries	0.67	2.45	0.64

Table 1: Tamil Grammar KG based Experimental Results

CES. It gauges users' perceptions of ease of use and the simplicity of completing tasks. A higher CES indicates that the users find the system easy to navigate and use. The highest Customer Effort Score calculated based on survey findings was 4.59.

3.2.2 Net Promoter Score (NPS)

NPS assesses the likelihood of users recommending the system or service to others. Figure 5b shows Audience Percentage Split in NPS. It is calculated based on user's responses to a single question: **How likely are you to recommend this system/service to a friend or colleague?** A higher NPS implies a higher chance an existing user shares the application to others. The highest Net Promoter Score calculated based on survey findings was 4.56.

3.2.3 Responsiveness

Responsiveness measures the system's ability to promptly address user queries, requests, or issues. Figure 5c shows Audience Percentage Split in Responsiveness. It evaluates the speed and efficiency with which the system handles user interactions, providing timely responses and assistance. A high level of responsiveness enhances user satisfaction by minimizing wait times. The Responsiveness Score calculated based on survey findings was 3.81.

3.2.4 Relevance

Relevance assesses the alignment between user's needs or preferences and the content or information provided by the system. Figure 5d shows Audience Percentage Split with respect to Relevance Score.

It evaluates the accuracy and appropriateness of the system's responses to user queries, ensuring that the information presented is useful to users. The highest Relevance Score calculated based on survey findings was 4.709.

3.2.5 Overall User Experience with *Aganittyam*

Overall Experience provides an aggregate measure of user's satisfaction with the system across various dimensions. Figure 5e shows Audience Percentage Split with respect to Relevance Score. It encompasses user's perceptions of usability, effectiveness, reliability, and satisfaction with the overall interaction. The highest Overall Experience Score calculated based on survey findings was 4.68.

4 Related Work

Though tools like Duolingo (Chen et al., 2020a) and Babbel (Hao et al., 2021) exists for language learning, they lack is serious pitfalls including limited depth, repetitive content, inconsistent quality across languages and lack of gamification over learning. Constructing a knowledge graph and querying using templatized questions is an emerging research in various real world domains (Ehrlinger and Wöß, 2016; Chen et al., 2018; Wu et al., 2019). Using annotated KGs constructed, Question-Answering systems are designed using various techniques, for example, via Automated Template Generation (Abujabal et al., 2017), using Knowledge Base Embeddings (Saxena et al., 2020) and so on. Few works for QA in Tamil are focused using Deep Learning Models (Murugathas and Thayasivam, 2022; Antony and Paul, 2022; Krishnan et al., 2023b) as mentioned in Section 1.1. Moreover, we observed

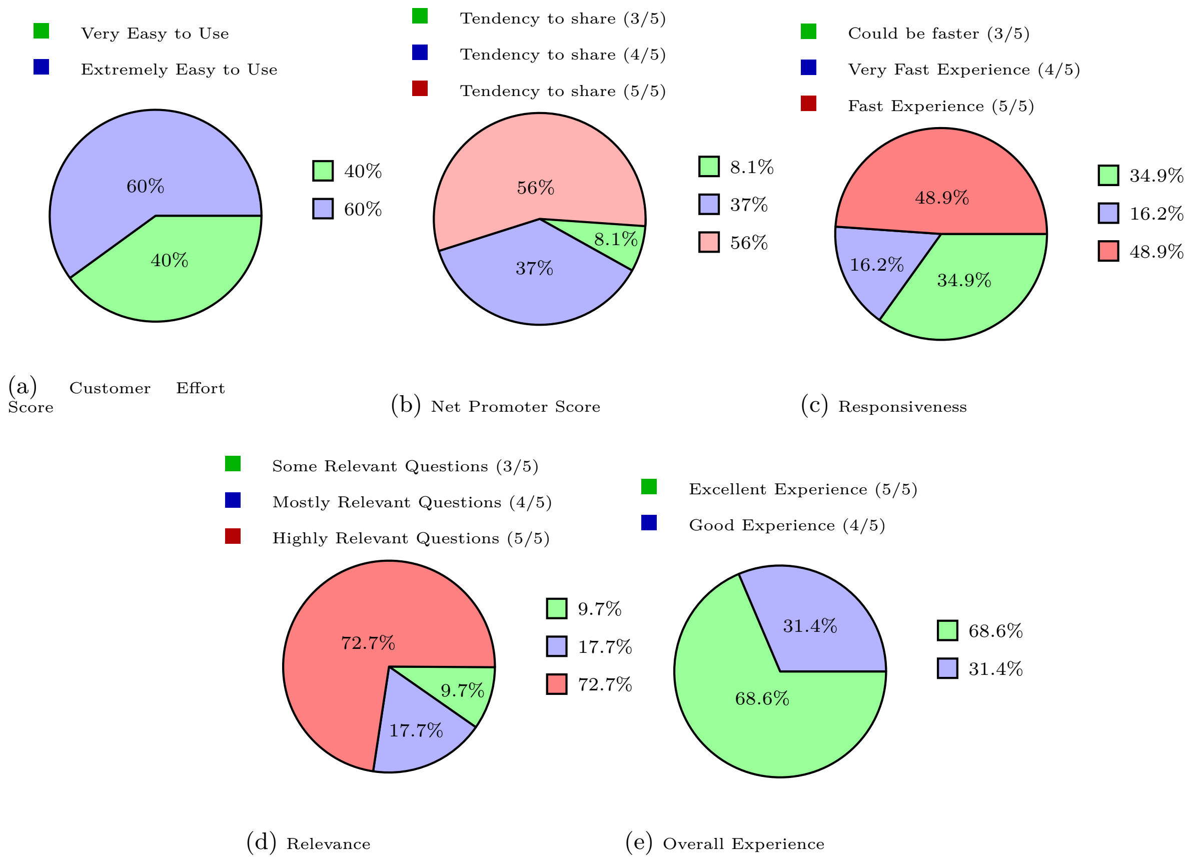


Figure 5: User Satisfaction metrics of Aganittyam

that to the best of our knowledge there is no such KG for Tamil Grammar for QA generation using common words used in everyday life. In this direction, a noteworthy effort in QA for Ramayana and Mahabharata includes a framework design for factoid question-answering in Sanskrit through automated construction of KGs for only human relationships (Terdalkar and Bhattacharya, 2023) and also a tool for annotating and querying KGs (Terdalkar and Bhattacharya, 2021). We refer the interested readers to Appendix A for a detailed survey of KG and QA systems.

5 Conclusions and Future Directions

In this work, we have presented *Aganittyam*, a novel Tamil grammar question-answering system that leverages knowledge graphs to facilitate learning and assessment of Tamil grammar. Our comprehensive corpus is designed to enable interactive learning experiences, with techniques that allow for automatic question-answering. The framework supports template-based question answering, providing a structured approach to learning. Our evaluation results demonstrate the robustness, reliability, and engaging nature of our QA system in answering various objective questions. Human-centric assessments indicate that the system is well-received by users. Currently, the knowledge graph includes basic grammar; however, we plan to enhance it with complex grammar types, poems, and stories in Tamil and other Dravidian languages.

Future research directions include expanding the knowledge graph to cover more topics and linguistic features, integrating additional question-answering techniques, and developing a mobile app version of *Aganittyam*. We also aim to address composition and complex question-answering for the grammar corpus and conduct further user studies to refine the system's usability and effectiveness.

Overall, our work illustrates the potential of knowledge graphs in facilitating language learning, with significant implications for the development of language education resources. The system is available at `https://aganittyam-web.onrender.com/home`, and we believe it can serve as a valuable tool for learners and educators alike.

References

Abdalghani Abujabal, Mohamed Yahya, Mirek Riedewald, and Gerhard Weikum. 2017. Automated template generation for question answering over knowledge graphs. In *Proceedings of the 26th international conference on world wide web*, pages 1191–1200.

Agattiyam Wikipedia. https://en.wikipedia.org /wiki/Agattiyam.

Betina Antony and NR Rejin Paul. 2022. Question answering system for tamil using deep learning. In *International Conference on Speech and Language Technologies for Low-resource Languages*, pages 244–252. Springer.

Ronald E Asher. 1985. *Tamil*, volume 7. Croom Helm London.

Byjus Page for TN Books. https://byjus.com .

Nilesh Chakraborty, Denis Lukovnikov, Gaurav Maheshwari, Priyansh Trivedi, Jens Lehmann, and Asja Fischer. 2019. Introduction to neural network based approaches for question answering over knowledge graphs. *arXiv preprint arXiv:1907.09361*.

Penghe Chen, Yu Lu, Vincent W Zheng, Xiyang Chen, and Xiaoqing Li. 2018. An automatic knowledge graph construction system for k-12 education. In *Proceedings of the fifth annual ACM conference on learning at scale*, pages 1–4.

Xiaojun Chen, Shengbin Jia, and Yang Xiang. 2020a. A review: Knowledge reasoning over knowledge graph. *Expert systems with applications*, 141:112948.

Zhe Chen, Yuehan Wang, Bin Zhao, Jing Cheng, Xin Zhao, and Zongtao Duan. 2020b. Knowledge graph completion: A review. *Ieee Access*, 8:192435–192456.

S. Choudhury et al. 2017. What do we really need for recurrent neural network training? *Neural Computation*, 29(11):2926–2954.

Rajarshi Das, Tsendsuren Munkhdalai, Xingdi Yuan, Adam Trischler, and Andrew McCallum. 2018. Building dynamic knowledge graphs from text using machine reading comprehension. *arXiv preprint arXiv:1810.05682*.

Velliangiri Dhanalakshmi, M Anand Kumar, RU Rekha, KP Soman, and S Rajendran. 2010. Grammar teaching tools for tamil language. In *2010 International Conference on Technology for Education*, pages 85–88. IEEE.

Lisa Ehrlinger and Wolfram Wöß. 2016. Towards a definition of knowledge graphs. *SEMANTiCS (Posters, Demos, SuCCESS)*, 48(1-4):2.

Dominique Estival, Chris Nowak, and Andrew Zschorn. 2004. Towards ontology-based natural language processing. In *Proceeedings of the Workshop on NLP and XML (NLPXML-2004): RDF/RDFS and OWL in Language Technology*, pages 59–66.

Dieter Fensel, Umutcan Simsek, Kevin Angele, Elwin Huaman, Elias Kärle, Oleksandra Panasiuk, Ioan Toma, Jürgen Umbrich, and Alexander Wahler. 2020a. *Knowledge graphs*. Springer.

Dieter Fensel, Umutcan Şimşek, Kevin Angele, Elwin Huaman, Elias Kärle, Oleksandra Panasiuk, Ioan Toma, Jürgen Umbrich, Alexander Wahler, Dieter Fensel, et al. 2020b. Introduction: what is a knowledge graph? *Knowledge graphs: Methodology, tools and selected use cases*, pages 1–10.

Giuseppe Futia and Antonio Vetrò. 2020. On the integration of knowledge graphs into deep learning models for a more comprehensible ai–three challenges for future research. *Information*, 11(2):122.

Nicola Guarino, Daniel Oberle, and Steffen Staab. 2009. What is an ontology? *Handbook on ontologies*, pages 1–17.

Xuejie Hao, Zheng Ji, Xiuhong Li, Lizeyan Yin, Lu Liu, Meiying Sun, Qiang Liu, and Rongjin Yang. 2021. Construction and application of a knowledge graph. *Remote Sensing*, 13(13):2511.

Ben Hixon, Peter Clark, and Hannaneh Hajishirzi. 2015. Learning knowledge graphs for question answering through conversational dialog. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 851–861.

Zhisheng Huang, Jie Yang, Frank van Harmelen, and Qing Hu. 2017. Constructing knowledge graphs of depression. In *Health Information Science: 6th International Conference, HIS 2017, Moscow, Russia, October 7-9, 2017, Proceedings 6*, pages 149–161. Springer.

S. Ji, X. Liu, Y. Wang, and X. Yang. 2021. A survey on deep learning for big data. *IEEE Access*, 9:21691–21715.

Mayank Kejriwal, Craig A Knoblock, and Pedro Szekely. 2021. *Knowledge graphs: Fundamentals, techniques, and applications*. MIT Press.

KG. https://images.app.goo.gl /7WjT8aFoWWkj4NAu7.

Aravind Krishnan, Srinivasa Ramanujan Sriram, Balaji Vishnu Raj Ganesan, and S. Sridhar. 2023a. An extractive question answering system for the tamil language. In *Proceedings: IoT, Cloud and Data Science*, volume 124 of *Advances in Science and Technology*, pages 312–319. Trans Tech Publications Ltd.

Aravind Krishnan, Srinivasa Ramanujan Sriram, Balaji Vishnu Raj Ganesan, and S Sridhar. 2023b. An extractive question answering system for the tamil language. *Advances in Science and Technology*, 124:312–319.

Vanessa Lopez, Pierpaolo Tommasi, Spyros Kotoulas, and Jiewen Wu. 2016. Queriodali: question answering over dynamic and linked knowledge graphs. In *The Semantic Web–ISWC 2016: 15th International Semantic Web Conference, Kobe, Japan, October 17–21, 2016, Proceedings, Part II 15*, pages 363–382. Springer.

Denis Lukovnikov, Asja Fischer, Jens Lehmann, and Sören Auer. 2017. Neural network-based question answering over knowledge graphs on word and character level. In *Proceedings of the 26th international conference on World Wide Web*, pages 1211–1220.

Rubika Murugathas and Uthayasanker Thayasivam. 2022. Domain specific question & answer generation in tamil. In *2022 International Conference on Asian Language Processing (IALP)*, pages 323–328.

Jay Pujara and Sameer Singh. 2018. Mining knowledge graphs from text. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, pages 789–790.

S Rajendran, M Anand Kumar, Ratnavel Rajalakshmi, V Dhanalakshmi, P Balasubramanian, and KP Soman. 2022. Tamil nlp technologies: Challenges, state of the art, trends and future scope. In *International Conference on Speech and Language Technologies for Low-resource Languages*, pages 73–98. Springer.

Amrita Saha, Vardaan Pahuja, Mitesh Khapra, Karthik Sankaranarayanan, and Sarath Chandar. 2018. Complex sequential question answering: Towards learning to converse over linked question answer pairs with a knowledge graph. In *Proceedings of the AAAI conference on artificial intelligence*.

Kengatharaiyer Sarveswaran. 2024. Morphology and syntax of the tamil language. *arXiv preprint arXiv:2401.08367*.

Apoorv Saxena, Soumen Chakrabarti, and Partha Talukdar. 2021. Question answering over temporal knowledge graphs. *arXiv preprint arXiv:2106.01515*.

Apoorv Saxena, Aditay Tripathi, and Partha Talukdar. 2020. Improving multi-hop question answering over knowledge graphs using knowledge base embeddings. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pages 4498–4507.

Himanshu Singh and Rajiv Ratn Shah. 2022. *TamilNLP: low resource language processing*. Ph.D. thesis, IIIT-Delhi.

Sanford B Steever. 2018. Tamil and the dravidian languages. In *The world's major languages*, pages 653–671. Routledge.

Tamil Wikinaotinary. https://ta.wiktionary.org.

Tamil Wikipedia. https://tamil.wiki.

Hrishikesh Terdalkar and Arnab Bhattacharya. 2021. Sangrahaka: A tool for annotating and querying knowledge graphs. In *Proceedings of the 29th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, pages 1520–1524.

Hrishikesh Terdalkar and Arnab Bhattacharya. 2023. Framework for question-answering in sanskrit through automated construction of knowledge graphs. *arXiv preprint arXiv:2310.07848*.

Sanju Tiwari, Fatima N Al-Aswadi, and Devottam Gaurav. 2021. Recent trends in knowledge graphs: theory and practice. *Soft Computing*, 25:8337–8355.

Turing Institute. https://tamil.wiki.

Ruijie Wang, Meng Wang, Jun Liu, Siyu Yao, and Qinghua Zheng. 2018. Graph embedding based query construction over knowledge graphs. In *2018 IEEE International Conference on Big Knowledge (ICBK)*, pages 1–8. IEEE.

Xindong Wu, Jia Wu, Xiaoyi Fu, Jiachen Li, Peng Zhou, and Xu Jiang. 2019. Automatic knowledge graph construction: A report on the 2019 icdm/icbk contest. In *2019 IEEE International Conference on Data Mining (ICDM)*, pages 1540–1545. IEEE.

Weiguo Zheng, Jeffrey Xu Yu, Lei Zou, and Hong Cheng. 2018. Question answering over knowledge graphs: question understanding via template decomposition. *Proceedings of the VLDB Endowment*, 11(11):1373–1386.

Weiguo Zheng and Mei Zhang. 2019. Question answering over knowledge graphs via structural query patterns. *arXiv preprint arXiv:1910.09760*.

Cunchao Zhu, Muhao Chen, Changjun Fan, Guangquan Cheng, and Yan Zhang. 2021. Learning from history: Modeling temporal knowledge graphs with sequential copy-generation networks. In *Proceedings of the AAAI conference on artificial intelligence*, pages 4732–4740.

Yuchen Zhuang, Yue Yu, Kuan Wang, Haotian Sun, and Chao Zhang. 2024. Toolqa: A dataset for llm question answering with external tools. *Advances in Neural Information Processing Systems*, 36.

A Knowledge Graphs and Question Answering Systems

In this section we present the literature on online language tools, KGs, and QA systems.

A.1 Online Language Learning Platforms

While popular language learning applications like Duolingo (Chen et al., 2020a) and Babbel (Hao et al., 2021) offer a fun and accessible way to pick up conversational Tamil, they often fall short when it comes to in-depth grammar instruction. These platforms are designed to prioritize spoken fluency, focusing on building vocabulary and sentence structures for everyday communication. This conversational focus means they may not delve into the complexities of Tamil grammar rules, such as verb conjugations, case systems, or proper sentence structure.

A.2 Knowledge Graph

The various techniques on knowledge graph construction are provided in (Tiwari et al., 2021; Fensel et al., 2020a). In the recent past decades, KG construction is drawing attention in various research problems in Information Extraction from documents, Web etc. (Ji et al., 2021). Some of the interesting work includes building dynamic KGs from text (Das et al., 2018), Automatic KG construction (Pujara and Singh, 2018), Temporal KGs (Zhu et al., 2021) and including domain specific KGs (Huang et al., 2017) etc.

A.3 Query Template Generation from Natural Language Questions

Generating templates i.e. Structured Queries for Question Answering over Knowledge Graphs where input questions are simplified by various NLP techniques. Some of the work include Graph Embedding based Query Construction by (Wang et al., 2018), Automated Template generation (Abujabal et al., 2017) and Question Understanding via Template decomposition (Zheng et al., 2018).

A.4 Question Answering

Significant works on this direction spans across works of use of Neural Networks (Lukovnikov et al., 2017), QA over KG via Structured Query Patterns (Zheng and Zhang, 2019), Querying of Dynamic KG (Lopez et al., 2016; Choudhury et al., 2017) and Querying over Temporal KG (Saxena et al., 2021) etc. These modified forms of Knowledge Graphs have tried to address various types of simple queries. Works on Complex queries through KG include creating a corpus by Priyansh Trivedi et al. (Chakraborty et al., 2019), application specific (Wireless Sensor Networks) (Zhuang et al., 2024) and Complex Sequential QA (Saha et al., 2018). In an another thread, Conversational QA through KG has gained attractions as well (Hixon et al., 2015).

A.5 Question Answering for Indian Languages

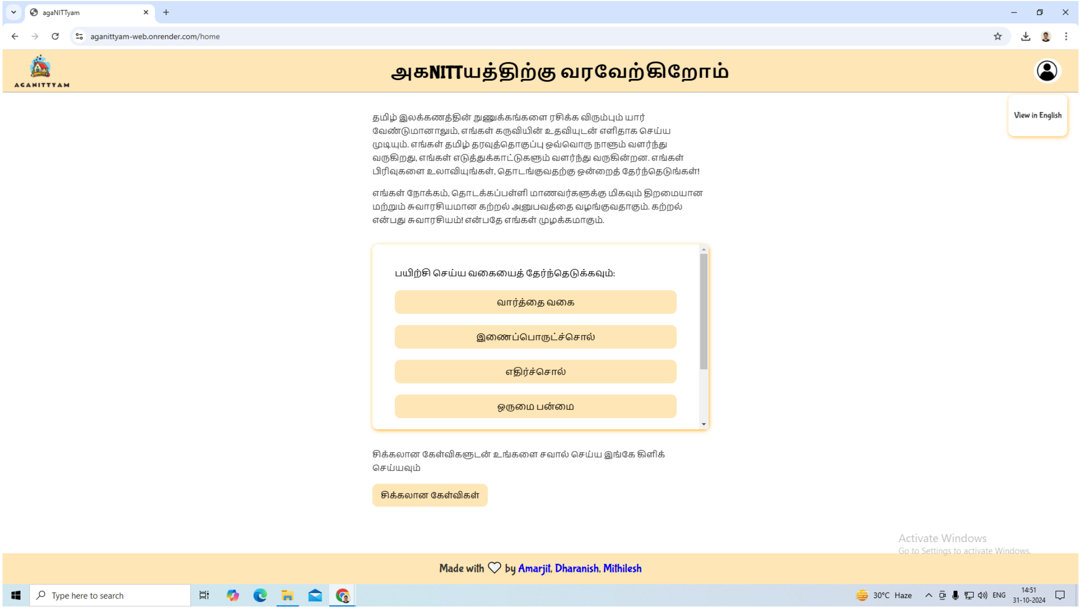
Notable effort in QA for Mahabharata and Ramayana includes a framework design for factoid Question Answering in Sanskrit through automated Construction of KGs (Terdalkar and Bhattacharya, 2023). This architecture is designed with multiple components and is developed based on user-defined rules and heuristics by incorporating Sanskrit language's grammar and its text structure. Another work by the same author includes the design of a web-based tool named 'Sangrahaka' for annotating entities and relationships and querying the KGs (Terdalkar and Bhattacharya, 2021). More details about 'Sangrahaka' is given in the subsequent subsection.

A.6 Sangrahaka: a Tool for annotating and querying Knowledge Graphs (Terdalkar and Bhattacharya, 2021)

Researchers have developed Sangrahaka, a web-based tool that empowers users to participate in the construction of these powerful Knowledge Graphs. Sangrahaka facilitates the annotation of textual corpora, enabling users to identify and link key entities within the text. In such applications, the Knowledge Graph serves as a pre-defined repository of knowledge, while Sangrahaka focuses on the initial stage of Knowledge Graph construction, where users actively contribute to the knowledge base through annotation. Sangrahaka functions like a digital highlighter for text documents. Users can pinpoint important entities, like people, places, or events, and then annotate the connections between them.

Existing Knowledge Graphs are pre-built with established rules while Sangrahaka focuses on users actively creating the Knowledge Graph by identifying and annotating elements in text sources. This makes the content user-driven and the tool versatile (works with various languages) and user-friendly. In Tamil grammar learning, the Knowledge Graph stores information about the language's building blocks – morphemes, parts of speech, and the rules governing their interaction. Anyone can then query this Knowledge Graph to gain a deeper understanding of fundamental Tamil grammar. This bridges the gap in current resources by providing a comprehensive and efficient way to master Tamil grammar's fundamentals. Motivated by *Sangrahaka* and other methods mentioned above, in this work, we present a novel framework by proposing a KG for Tamil Grammar for all types of learners by constructing an ontology about entity types and relationship and performing human annotations on the corpus. Subsequently, we developed a QA system for answering templatized queries and some complex queries.

B Aganittyam UI



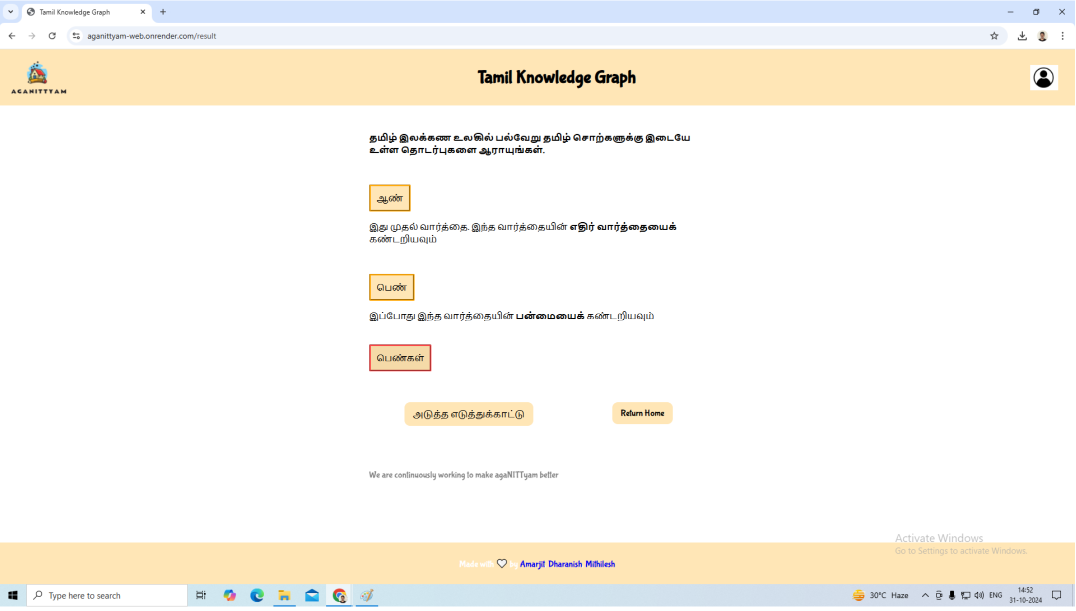
தமிழ் இலக்கணத்தின் நுணுக்கங்களை ரசிக்க விரும்பும் யார் வேண்டுமானாலும், எங்கள் கருவியின் உதவியுடன் எளிதாக செய்ய முடியும். எங்கள் தமிழ் தரவுத்தொகுப்பு ஒவ்வொரு நாளும் வளர்ந்து வருகிறது, எங்கள் எடுத்துக்காட்டுகளும் வளர்ந்து வருகின்றன. எங்கள் பிரிவுகளை உலாவியுங்கள், தொடங்குவதற்கு ஒன்றைத் தேர்ந்தெடுங்கள்!

எங்கள் நோக்கம், தொடக்கப்பள்ளி மாணவர்களுக்கு மிகவும் திறமையான மற்றும் சுவாரசியமான கற்றல் அனுபவத்தை வழங்குவதாகும். கற்றல் என்பது சுவாரசியம்! என்பதே எங்கள் முழக்கமாகும்.

பயிற்சி செய்ய வகையைத் தேர்ந்தெடுக்கவும்:

சிக்கலான கேள்விகளுடன் உங்களை சவால் செய்ய இங்கே கிளிக் செய்யவும்

Figure 6: Aganittyam UI



தமிழ் இலக்கண உலகில் பல்வேறு தமிழ் சொற்களுக்கு இடையே உள்ள தொடர்புகளை ஆராயுங்கள்.

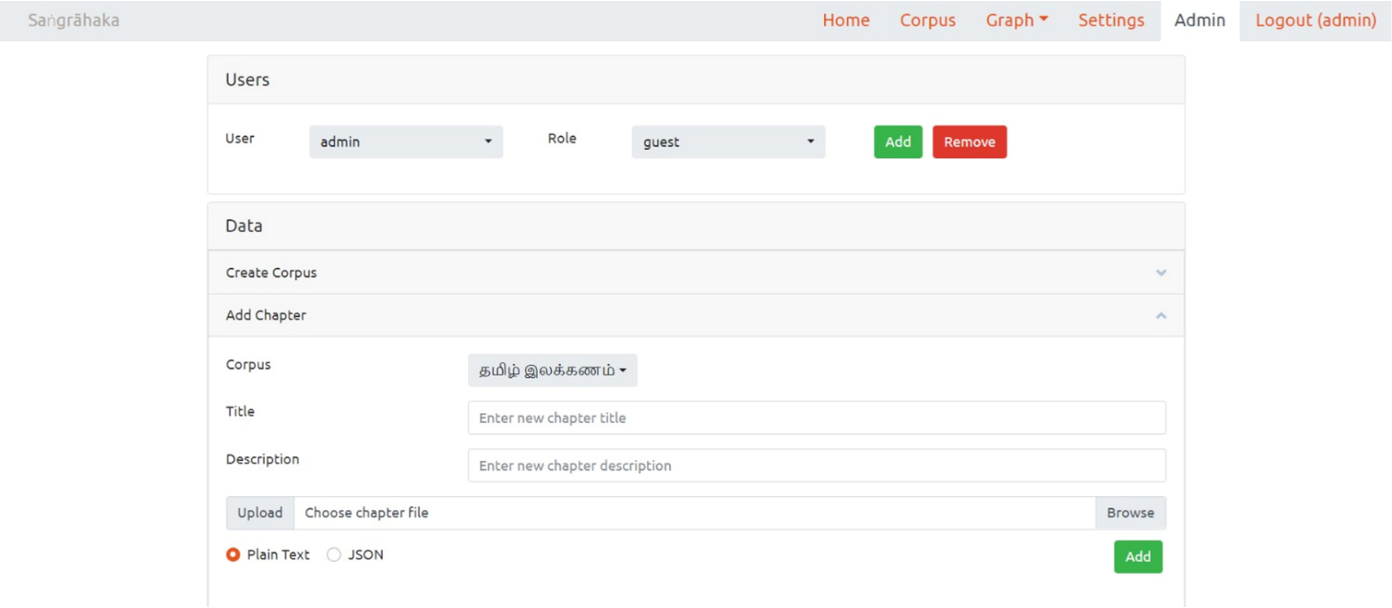
இது முதல் வார்த்தை. இந்த வார்த்தையின் **எதிர் வார்த்தையைக்** கண்டறியவும்

இப்போது இந்த வார்த்தையின் **பன்மையைக்** கண்டறியவும்

We are continuously working to make agaNITTyam better

Figure 7: Quiz Portal

C Sangrahaka UI



Users

User admin

Role guest

Data

Create Corpus

Add Chapter

Corpus தமிழ் இலக்கணம்

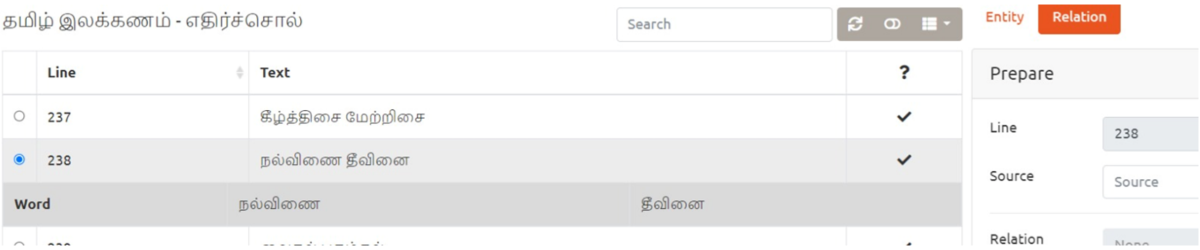
Title Enter new chapter title

Description Enter new chapter description

Upload Choose chapter file

● Plain Text ○ JSON

Figure 8: Creating the Corpus



தமிழ் இலக்கணம் - எதிர்ச்சொல்

	Line	Text	?
○	237	கீழ்த்திசை மேற்றிசை	✓
●	238	நல்விணை தீவினை	✓

Word	நல்விணை	தீவினை

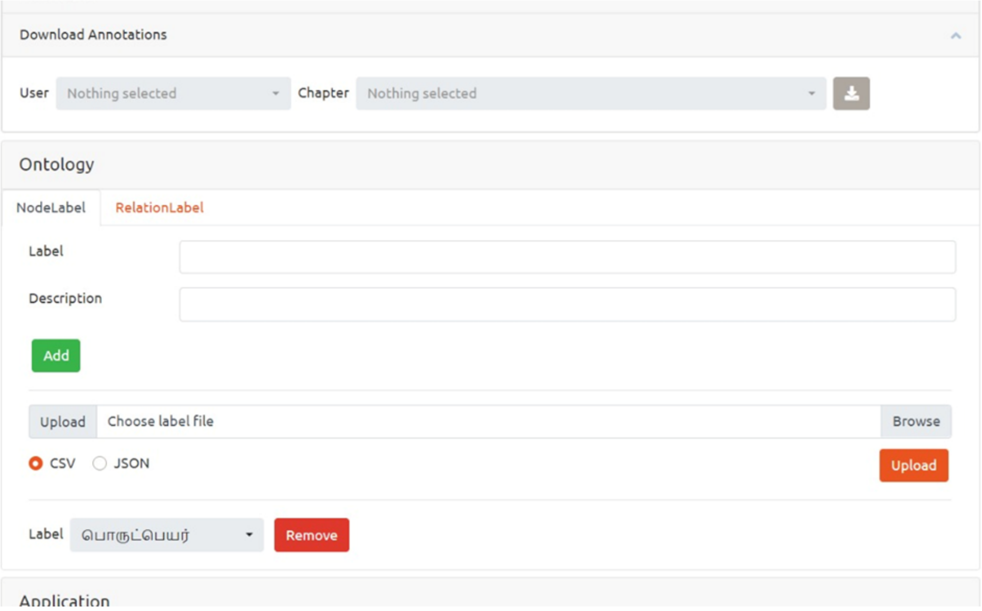
Prepare

Line 238

Source Source

Relation None

Figure 9: Adding Relation Types to the corpus.



Download Annotations

User Nothing selected
Chapter Nothing selected

Ontology

NodeLabel RelationLabel

Label

Description

Add

Upload Choose label file Browse

● CSV ○ JSON

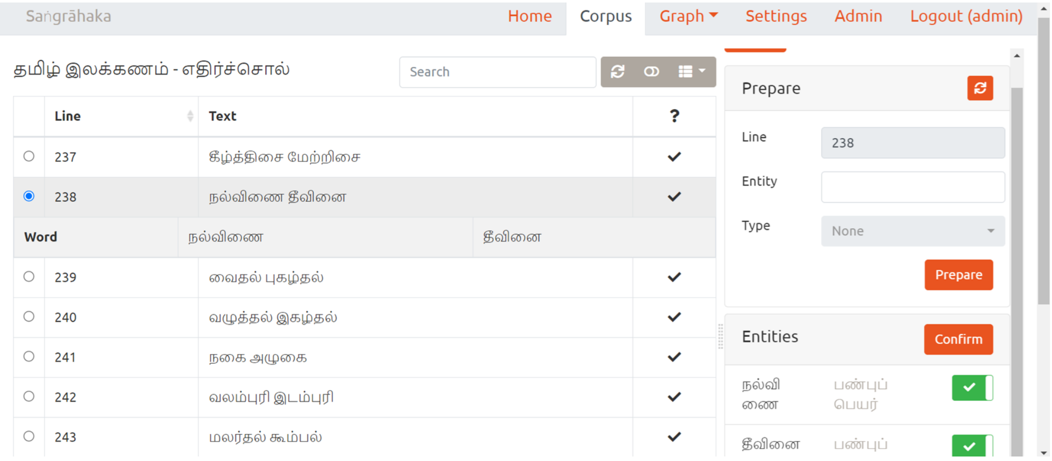
Upload

Label பொருட்பெயர்

Remove

Application

Figure 10: Creating Ontology



தமிழ் இலக்கணம் - எதிர்ச்சொல்

Search

	Line	Text	?
○	237	கீழ்த்திசை மேற்றிசை	✓
●	238	நல்விணை தீவினை	✓
Word	நல்விணை	தீவினை	
○	239	வைதல் புகழ்தல்	✓
○	240	வழுத்தல் இகழ்தல்	✓
○	241	நகை அழுகை	✓
○	242	வலம்புரி இடம்புரி	✓
○	243	மலர்தல் கூம்பல்	✓

Prepare

Line 238
Entity
Type None

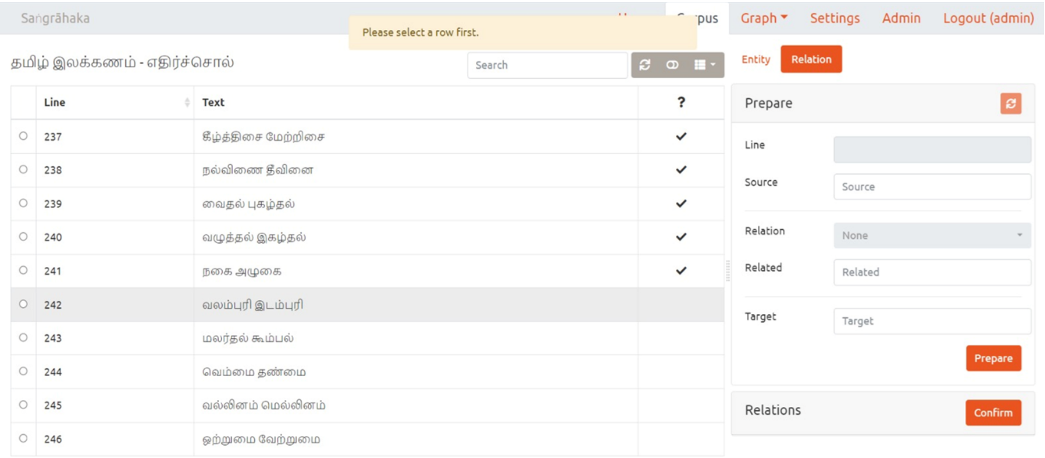
Prepare

Entities

Confirm

நல்விணை	பண்புப் பெயர்	✓
தீவினை	பண்புப்	✓

Figure 11: Annotation of Tamil Grammar Relation Types



Please select a row first.

தமிழ் இலக்கணம் - எதிர்ச்சொல்

Search

	Line	Text	?
○	237	கீழ்த்திசை மேற்றிசை	✓
○	238	நல்விணை தீவினை	✓
○	239	வைதல் புகழ்தல்	✓
○	240	வழுத்தல் இகழ்தல்	✓
○	241	நகை அழுகை	✓
○	242	வலம்புரி இடம்புரி	
○	243	மலர்தல் கூம்பல்	
○	244	வெம்மை தண்மை	
○	245	வல்லினம் மெல்லினம்	
○	246	ஒற்றுமை வேற்றுமை	

Entity Relation

Prepare

Line
Source Source
Relation None
Related Related
Target Target

Prepare

Relations

Confirm

Figure 12: Labelling Features