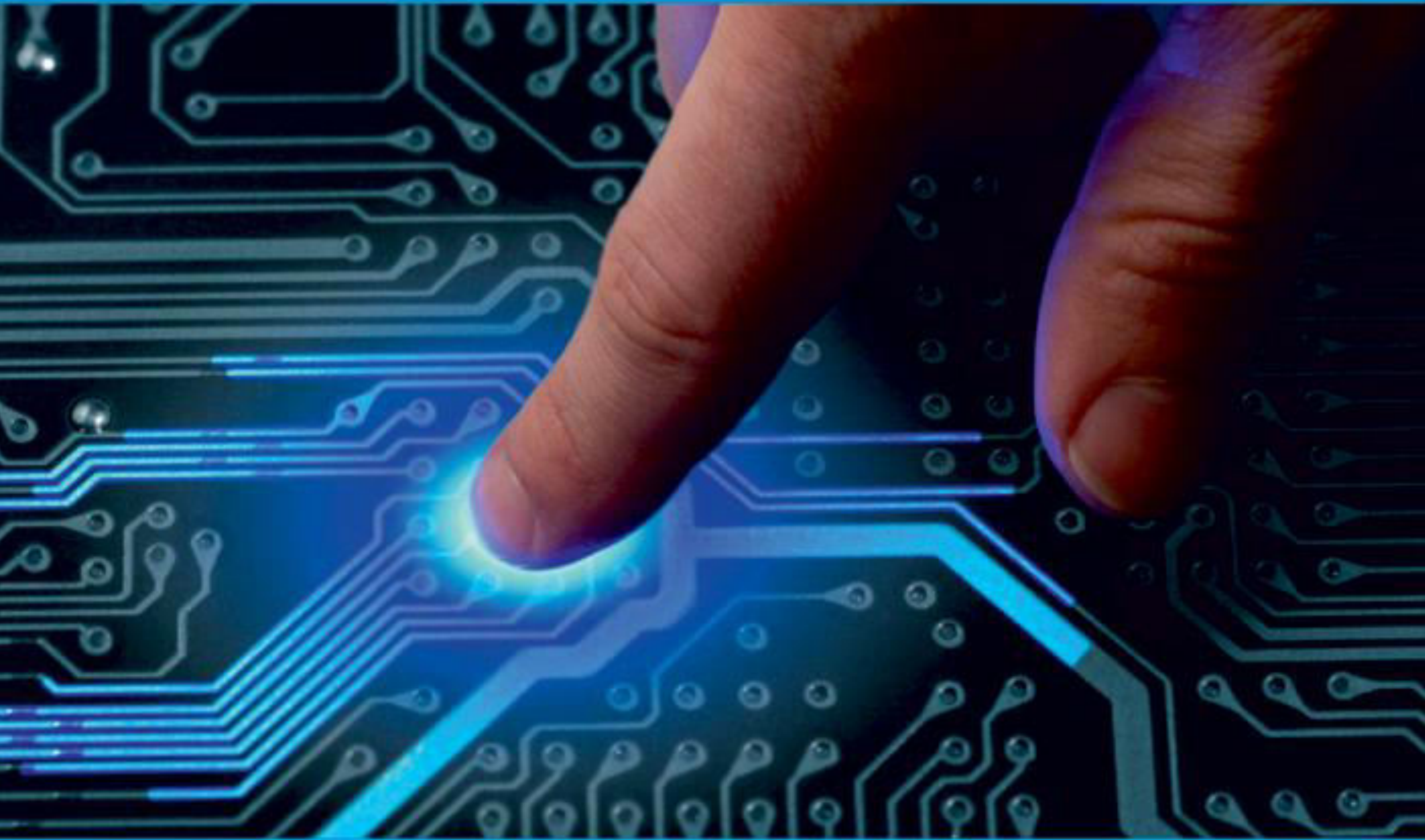
IJIRCCE

e-ISSN: 2320-9801 | p-ISSN: 2320-9798



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

Volume 12, Issue 11, November 2024

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

Impact Factor: 8.625

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.625| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Open Domain Tamil Question Answering System

Niveditha S[1], Arnav Ajay Krishna[2], Aditilakshmi S[3]

Assistant Professor, Department of C. S. E, SRM Institute of Science and Technology, Vadapalani, Chennai, India[1]

B. Tech Student, Department of C.S. E, SRM Institute of Science and Technology, Vadapalani, Chennai, India[2]

B. Tech Student, Department of C.S. E, SRM Institute of Science and Technology, Vadapalani, Chennai, India[3]

ABSTRACT: The Open Domain Tamil Question Answering System is an initiative that endeavours to furnish precise and contextually pertinent responses to a broad spectrum of inquiries in the Tamil language. We built the open domain Question Answering system using web scraping methodologies. It tokenizes Tamil words using the Tamil corpus. Once tokenized, it is sent for information retrieval. The system ensures efficient and precise information retrieval by integrating natural language processing techniques for lemmatization and morphological analysis with Bidirectional Encoder Representations from Transformer (BERT) Algorithms. This system offers a thorough approach to information retrieval that yields precise and manageable responses.

KEYWORDS: Tamil Question Answering, Natural Language Processing, Transformer Models, Wikipedia API, Tamil Language, Open Domain QA, BERT, Morphological Processing

I. INTRODUCTION

Tamil is one of the oldest and morphologically rich languages.[4] Despite its rich literary tradition, advanced tools for retrieving information in Tamil are scarce. Although there are various available resources to retrieve information in Tamil, by providing a concise and to the point answer, the entire process becomes more user friendly.

Current methods are built on top of Named Entity Recognition (NER) models. However, they are less effective in Tamil compared to those in English, making it challenging to correctly identify and classify proper nouns, dates, and other critical information needed for answering questions. The other common issue is that most of them are often domain dependent, resulting in a high limitation in terms of usage. Although there are a few tools for open domain Tamil question answering, due to extensive grammar and lack of support in language processing for Tamil, they are inaccurate and requires a lot of work. Another common issue with some of these models are that they are built on static data. Static QA systems rely on the data they were trained on, which can become outdated quickly, especially in domains in which information changes rapidly.

With all this information, this paper introduces a novel approach to developing an Open Domain Tamil Question Answering (QA) system. By integrating the Wikipedia API and advanced NLP techniques, the system is designed to handle a wide range of queries and provide contextually accurate answers. The proposed system not only bridges the information gap but also paves the way for improved access to technology for Tamil speakers.

II. LITERATURE REVIEW

The development of question-answering systems has advanced significantly with models like ELECTRA, which introduced Replaced Token Detection as an alternative to Masked Language Modeling (MLM).[1] By corrupting input data with token replacements and training the model to detect these alterations, ELECTRA improves efficiency in high-resource languages, outperforming traditional MLM models. However, despite its impressive performance, the model requires extensive fine-tuning when applied to low-resource languages such as Tamil.

In contrast, the study on Human-Computer Text Conversation through NLP in Tamil using Intent Recognition demonstrates a more targeted application for Tamil. This project created a chatbot system to help parents track their children's academic progress by using Natural Language Processing (NLP) to interpret and respond to queries in Tamil.[2] The system successfully bridges communication gaps between parents and educational institutions, emphasizing the potential of Tamil language systems in practical applications. However, this system is limited to its domain, and the initial setup can be complex, requiring careful consideration for deployment.

Another approach to question-answering is the Domain Knowledge Enriched Framework for Restricted Domain Question Answering Systems, which integrates domain-specific knowledge with NLP techniques. This framework improves accuracy by focusing on specialized fields and using predefined domain rules.[3] While this approach is highly effective within restricted domains, it suffers from scalability issues when applied to broader open-domain contexts. The reliance on domain-specific knowledge makes it difficult for the system to adapt to other areas without significant reconfiguration, limiting its flexibility and general applicability.

Jacob Devlin et al., in their paper "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding" (2018),[6] introduced BERT as a model designed to pre-train bidirectional representations from text by conditioning on both left and right context at all layers. Unlike prior models that were limited by unidirectional context (such as GPT), BERT overcomes these limitations through its masked language model (MLM) and next sentence prediction (NSP) tasks, making it more effective for a wide range of natural language processing tasks, including question answering. BERT has demonstrated state-of-the-art performance across multiple NLP benchmarks, including the Stanford Question Answering Dataset (SQuAD), making it particularly powerful for tasks that require deep understanding of context from multiple directions.

After analyzing several papers, this system architecture was devised. It is an extensive system that leverages the WikipediaAPI (Tamil) as a tool that serves as a dynamic information source. It additionally adopts NLP techniques and creates Tamil morphemes and Lemmatization on Tamil words. Using linguistic papers as reference, the system was designed to filter out stop words appropriately. Finally, using BERT, it facilitates appropriate answer generation in Tamil, providing the required results.

III. PROPOSED SYSTEM

A. Proposed methodology

The proposed Open Domain Tamil Question Answering System aims to bridge the language gap in information access for Tamil users. The system uses the Wikipedia API to fetch the latest articles relevant to the question. It preprocesses the question by removing Tamil stop-words[5] and deriving root words, ensuring that the query focuses on meaningful content. Consequently, it searches for paragraphs within the retrieved articles that contain the derived root words and other significant words from the question, scoring and selecting the paragraph that best matches the query. After identifying the most relevant paragraph, a Hugging Face model is deployed to summarize the content, generating a concise and accurate answer for the user.

B. Proposed architecture:

The process begins with an input, namely, user questions. The questions are typically expressed in natural language, and the articles serve as the source of information from which the system will extract the most relevant answers. The objective is to efficiently retrieve relevant information from the Tamil articles that directly addresses the user's queries.

Once the question is provided, the system moves into the preprocessing stage. Preprocessing involves cleaning and normalizing both the questions and the Tamil articles. For the questions, this includes tasks such as removing stop-words[5], converting the text to lowercase, or correcting spelling errors. Upon basic preprocessing, the system tokenizes the inputs and the context is derived. Tokenization breaks down the question into smaller, more manageable components called tokens. These tokens are typically words or subwords. By splitting the text into tokens, the system can more easily interpret and manipulate the individual parts of the text, facilitating accurate understanding at a granular level.

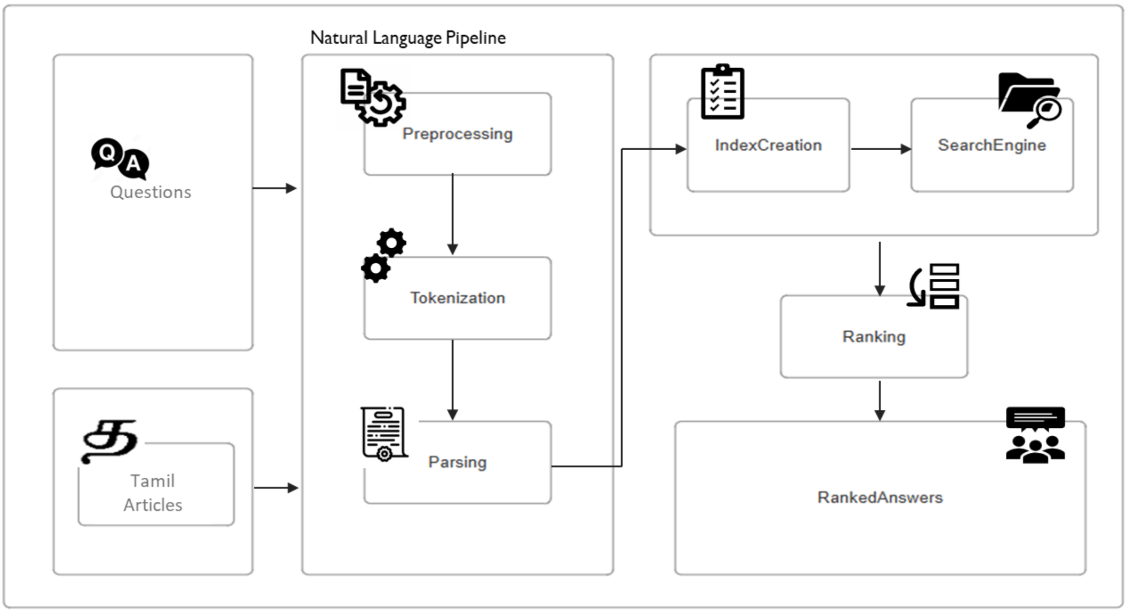


FIG 3.1: ARCHITECTURE DIAGRAM OF THE MODEL

Following tokenization, the system proceeds with parsing. Parsing is the process of analyzing the grammatical structure of the tokenized input. By identifying the relationships between different tokens—such as the roles of subjects, verbs, and objects—the system ensures that both the questions and the Tamil articles are comprehensively understood. This step is critical for determining the correct meaning and context, preparing the text for the next stage of the pipeline.

Once the text has been parsed, the relevant words are searched using the Wikipedia API. It searches not only the article titles but also the paragraph it might be of interest. After this, using the model developed using BERT, an index of the relevant Tamil articles is created. Index creation is a crucial step that allows the system to organize the articles in a searchable format. This index functions as a structured map of the articles, enabling the search engine to efficiently retrieve relevant sections of text when a query is made.

At this stage, the search engine is activated to find relevant answers based on the user's query. Using the preprocessed, tokenized, and parsed question, the search engine queries the indexed Tamil articles. The search engine is responsible for matching the question with potential answers within the articles, using algorithms designed to locate the most relevant sections of text. This step narrows down the vast amount of data to the most pertinent information that may answer the user's question.

After retrieving relevant information, the system moves into the ranking phase where the appropriate context for the question is ranked better. The retrieved answers are not presented in raw form; instead, they are ranked based on their relevance to the original question. The ranking process ensures that the best and most accurate answers are prioritized. Various factors, such as keyword matching and semantic relevance, are used to determine the rank of each answer, providing the user with the most useful information at the top of the list.

In the final stage, the system delivers the ranked answers. These are the sections from the Tamil articles that have been identified as the most relevant to the user's question, ordered based on their ranking. By presenting the information in this way, the system allows the user to quickly find the most pertinent answers, ensuring that the output is both concise and accurate. The flow from input to output ensures a streamlined process, combining natural language processing with efficient information retrieval techniques.

C. Algorithm Used:

Step 1: User Query Input: Capture the user's query in Tamil and preprocess the text by normalizing, removing stop-words, and deriving root words using Tamil-specific text processing techniques.

Step 2: Document Retrieval: Formulate a search query using the processed root words and retrieve relevant documents or articles from Wikipedia using the Wikipedia API. Parse the retrieved documents into individual paragraphs for further analysis.

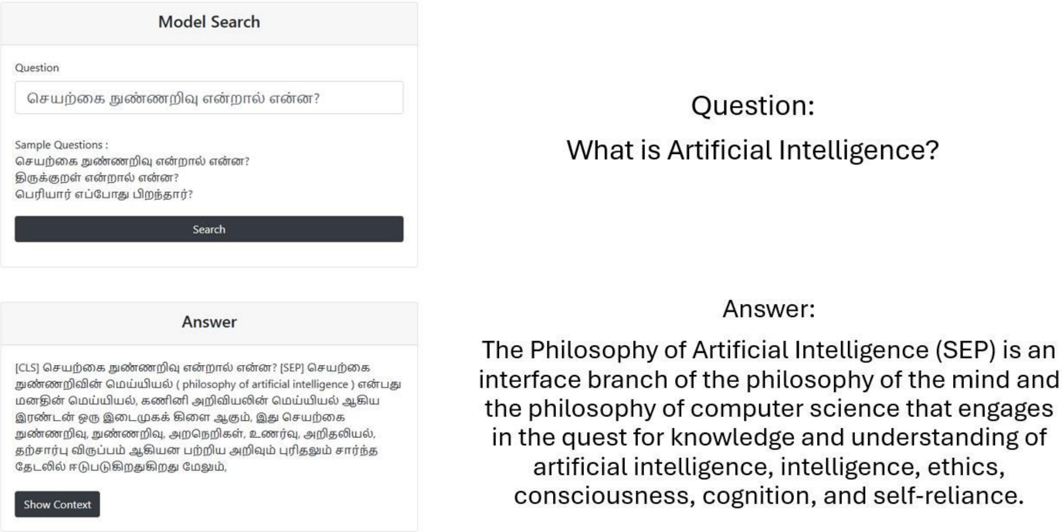
Step 3: Feature Extraction: Tokenize the text into words or n-grams, perform syntactic parsing to understand the grammatical relationships, and match the query tokens with those in the retrieved paragraphs, focusing on root words and syntactic relevance.

Step 4: Relevance Ranking: Calculate a relevance score for each paragraph based on token matching and context, using metrics like TF-IDF or cosine similarity, and rank the paragraphs according to their relevance to the query.

Step 5: Output Delivery: Present the summarized answer to the user in a clear and contextually accurate format, ensuring the response meets the query's intent.

IV. RESULTS

The system efficiently retrieves relevant information from Tamil articles via the Wikipedia API. It demonstrates an average response time of 6 seconds per query. This response time is competitive with other models, given the complexity of the Tamil language and open-domain nature of the system. This is due to the preprocessing steps, including tokenization, stop-word removal, and lemmatization being performed swiftly and accurately, focusing on the most meaningful aspects of the query. This allows for almost real-time interaction for the user, which increases the usability of the system.



Model Search

Question

செயற்கை நுண்ணறிவு என்றால் என்ன?

Sample Questions :
செயற்கை நுண்ணறிவு என்றால் என்ன?
திருக்குறள் என்றால் என்ன?
பெரியார் எப்போது பிறந்தார்?

Answer

[CLS] செயற்கை நுண்ணறிவு என்றால் என்ன? [SEP] செயற்கை நுண்ணறிவின் மெய்யியல் (philosophy of artificial intelligence) என்பது மனதின் மெய்யியல், கணினி அறிவியலின் மெய்யியல் ஆகிய இரண்டன் ஒரு இடைமுகக் கிளை ஆகும், இது செயற்கை நுண்ணறிவு, நுண்ணறிவு, அறநெறிகள், உணர்வு, அறிதலியல், தற்சார்பு விருப்பம் ஆகியன பற்றிய அறிவும் புரிதலும் சார்ந்த தேடலில் ஈடுபடுகிறதுகிறது மேலும்,

Question:

What is Artificial Intelligence?

Answer:

The Philosophy of Artificial Intelligence (SEP) is an interface branch of the philosophy of the mind and the philosophy of computer science that engages in the quest for knowledge and understanding of artificial intelligence, intelligence, ethics, consciousness, cognition, and self-reliance.

Fig 4.1: Sample Output from the Model Built

One of the key strengths of the developed system is, of course, the open-domain model. This allows it to handle queries from a wide variety of topics without being restricted to a specific subject matter. The system is not bound by static training data, which is a common issue seen in traditional QA systems. The Wikipedia API supplies relevant and up to date information, making sure the system is pulling data from a continuously expanding repository of articles.

The integration of the BERT model significantly enhanced the system's ability to retrieve relevant answers. The bidirectional approach used by BERT allowed for better understanding of the syntax and semantics of Tamil sentences. Complex sentences, such as questions with multiple clauses or ambiguous phrasing benefited from BERT's deep language modelling, which improved answer relevance. This also reduced the error rate for sentences containing homonyms(words with multiple meanings), due to the BERT approach. When the system parses complex, nuanced questions where multiple possible answers exist in the same paragraph, it encounters some issues, so there may be further room for improvement in this module.

V. CONCLUSION AND FUTURE WORK

In this paper, we presented an Open Domain Tamil Question Answering System, designed to handle the complexities of Tamil language processing for a wide range of queries. By leveraging BERT-based models and integrating natural language processing techniques specific to Tamil, the system demonstrated an improved capacity for understanding and retrieving answers in a low-resource language. Additionally, the system's architecture incorporates a multi-stage pipeline that includes preprocessing, tokenization, parsing, and index creation to handle large volumes of Tamil text efficiently. The integration of a ranking mechanism ensures that the most relevant answers are prioritized based on semantic relevance and keyword matching.

However, certain limitations remain. For example, in queries like " ‘நான்’ என்றால் என்ன?", which means "What does ‘நான்’ mean?", the system struggles to retrieve relevant context, and in cases where Tamil words have multiple meanings that are closer to each other (Example: மரம், நாள்), incorrect contexts are sometimes returned. These challenges highlight the need for further improvements in context retrieval and disambiguation for Tamil.

In future work, we propose building a comprehensive Tamil knowledge graph that can better handle the semantic relationships between words and their meanings in different contexts. This graph will enable the system to distinguish between multiple meanings of the same word, providing more accurate answers by associating queries with the correct context. Additionally, the unique morphological structure and stemming patterns in Tamil provide a new realm of possibilities for developing a BERT-like model specifically tailored for the language. The rich inflectional system of Tamil, along with its agglutinative nature, offers opportunities to capture deeper linguistic nuances.

REFERENCES

1. Kevin Clark, Minh-Thang Luong, Quoc V. Le, Christopher D. Manning, (2020). ELECTRA: PRE-TRAINING TEXT ENCODERS AS DISCRIMINATORS RATHER THAN GENERATORS, The Eighth International Conference on Learning Representations (ICLR 2020)
2. Udhayakumar Shanmugam, Rajeswari P., Sowjanya Mani, Sneha Sivakumar, (2019). Human-Computer Text Conversation through NLP in Tamil using Intent Recognition, 2019 International Conference on Vision Towards Emerging Trends in Communication and Networking (ViTECoN)
3. Nidhi Malik, Aditi Sharan, Payal Biswas (2013). Domain Knowledge enriched framework for restricted domain question answering system, 2013 IEEE International Conference on Computational Intelligence and Computing Research
4. Betina Antony & NR Rejin Paul (2023). Tamil Question Answering System Using Machine Learning, First International Conference, SPELLL 2022, Kalavakkam, India, November 23–25. DOI:10.1007/978-3-031-33231-9_17
5. M.S. Faathima Fayaza & F. Fathima Farhath (2021). Towards Stopwords Identification in Tamil Text Clustering, International Journal of Advanced Computer Science and Applications, Vol. 12, No. 12, 2021
6. Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, Proceedings of NAACL-HLT 2019, pages 4171–4186. 2019 Association for Computational Linguistics

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING