JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

Prashnadyuti: An Automatic Multilingual QA Generating System

Dr. Nilesh Joshi[1], Ms. Leena Pande[2]

1. CFILT, IIT Bombay, joshinilesh60@gmail.com, 2. Maxwell Senior college, Nagpur, leenaspande@gmail.com

Abstract

This paper introduces ***Prashnadyuti***, our vision for a smart, multilingual system that automatically creates diverse educational quizzes. Imagine a tool that truly understands language! By harnessing the latest breakthroughs in Natural Language Processing (NLP) and machine learning (Vaswani et al., 2017), ***Prashnadyuti*** aims to meet the growing need for learning materials that are both personal and easy to access, no matter what language you speak. We'll walk you through how such a system could be built, explore its key parts, and show how it can generate all sorts of multiple-choice questions (MCQs)—from number puzzles to vocabulary challenges and factual checks—across many different languages. We'll wrap up by looking at the exciting challenges ahead and what's next for making multilingual question generation even better.

Keywords:

Automatic Question Generation (AQG), Artificial Intelligence in Education (AIED), Multilingual NLP, Distractor Generation, Large Language Models, Indic Language Computing, Educational Assessment.

1. Introduction

In our fast-paced, interconnected world, everyone deserves access to education that's flexible and understandable. Right now, creating good questions for tests and practice usually means a lot of manual effort. It's time-consuming, costly, and frankly, it limits how much variety and personalization we can offer in learning. But here's the exciting part: with amazing advancements in Natural Language Processing (NLP) and powerful large language models (Brown et al., 2020), we now have a fantastic opportunity to automate and supercharge this process!

That's why we're proposing "***Prashnadyuti***" (Sanskrit: □□□□□□□□□□□□). This beautiful Sanskrit word combines ***Prashna***, meaning "question," with ***Dyuti***, meaning "light" or "radiance"—so, it truly means the "light of questions." Our vision for ***Prashnadyuti*** is a clever, multilingual system that can automatically dig into text or data, pull out all the important bits, and then craft relevant questions and answers in multiple languages. Think of the possibilities! This system could be a game-changer for online learning platforms, language-learning apps, and anyone who needs dynamic educational content. It would empower educators and learners to quickly whip up customized quizzes and assessments. In the following sections, we'll dive into why ***Prashnadyuti*** is so important, how we imagine its architecture, and showcase its potential by highlighting its ability to generate the kinds of diverse quiz questions you've seen in our interactive MCQ webpage.

2. Background and Related Work

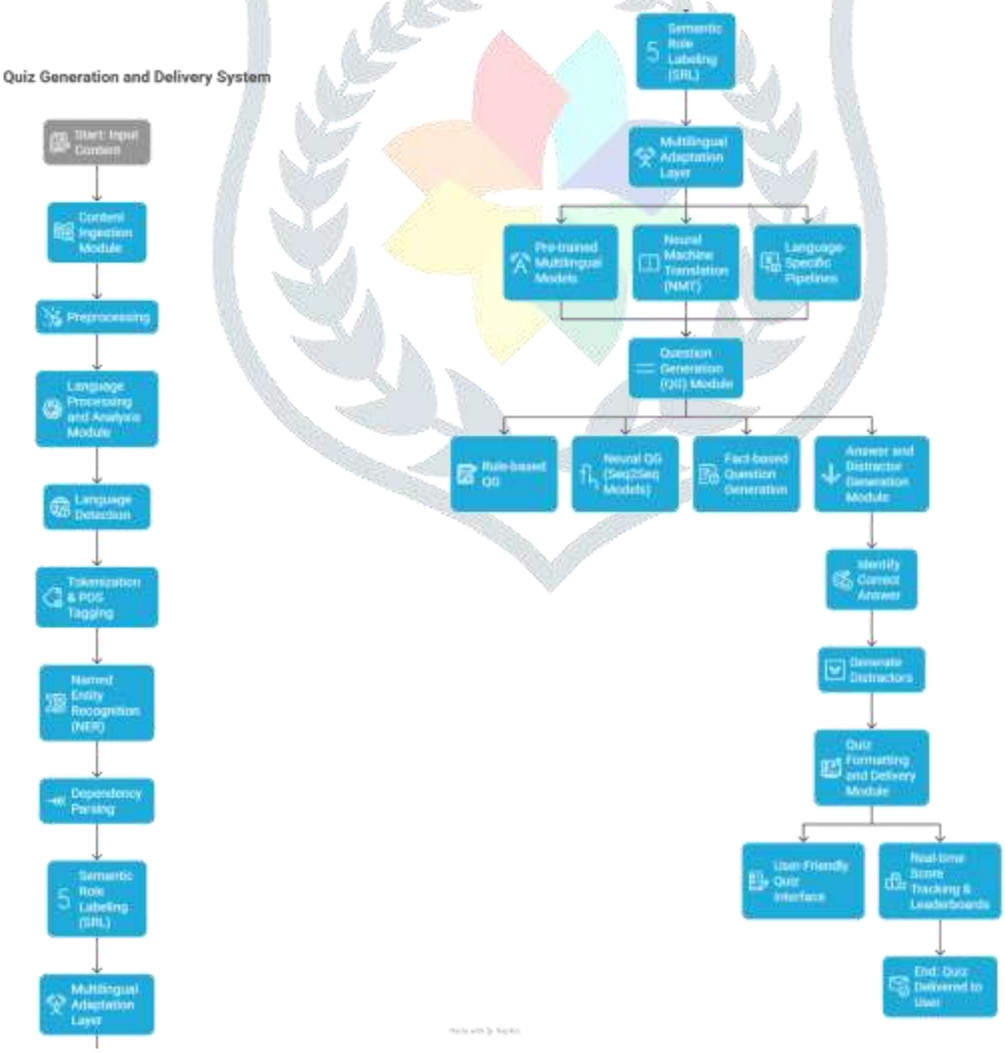
For a long time, researchers in NLP have been fascinated by Question Answering (QA) systems. We've come a long way from simple rule-based approaches to incredibly smart machine learning models (Jurafsky & Martin, 2009). Initially, QA systems were like super-efficient librarians, just pulling exact answers from structured databases. But now, with open-domain and generative QA systems, they're more like brilliant conversationalists, able to understand everyday language questions and create answers from mountains of unorganized text (Rajpurkar et al., 2018).

Making QA systems truly multilingual is a fascinating puzzle. It’s not just about translating words; it’s about grasping cultural nuances, handling different sentence structures, and adapting to various educational approaches. Often, we might translate a question or source text, or even build separate models for each language (Conneau et al., 2018). Then there's Question Generation (QG), which is like QA in reverse: it takes text and cooks up questions from it, often to help people understand what they've read or to create assessments. While we've made huge strides in QA and QG for widely spoken languages, building truly robust and flexible multilingual QG systems is still a thrilling frontier in research (Du & Cardie, 2017).

3. System Architecture of *Prashnadyuti*

We've designed ***Prashnadyuti*** to be super flexible, built with different parts that can easily work together. Here's a peek at how this intelligent system could be put together:

Here is a visual representation of the system's architecture using a flowchart:



```
graph TD
    A[Start: Input Content (Text/Data)] --> B{Content Ingestion Module};
    B --> C[Preprocessing];
    C --> D{Language Processing and Analysis Module};
    D --> D1[Language Detection];
    D1 --> D2[Tokenization & POS Tagging];
    D2 --> D3[Named Entity Recognition (NER)];
    D3 --> D4[Dependency Parsing];
    D4 --> D5[Semantic Role Labeling (SRL)];
    D5 --> E{Multilingual Adaptation Layer};
    E --> |If high-resource language| E1[Pre-trained Multilingual Models];
    E --> |If sparse language resources| E2[Neural Machine Translation (NMT)];
    E --> |For optimized processing| E3[Language-Specific Pipelines];
    E1 --> F{Question Generation (QG) Module};
    E2 --> F;
    E3 --> F;
    F --> F1[Rule-based QG];
    F --> F2[Neural QG (Seq2Seq Models)];
    F --> F3[Fact-based Question Generation];
    F --> G{Answer and Distractor Generation Module};
    G --> G1[Identify Correct Answer];
    G1 --> G2[Generate Distractors];
    G2 --> H{Quiz Formatting and Delivery Module};
    H --> H1[User-Friendly Quiz Interface (e.g., MCQ Webpage)];
    H --> H2[Real-time Score Tracking & Leaderboards (e.g., Firebase)];
    H2 --> I[End: Quiz Delivered to User];
```

3.1. Content Ingestion Module

This is where the magic begins! This module grabs all kinds of information—from simple text documents and web pages to detailed academic papers or structured data. For our educational goals, this could mean anything from chapters in a textbook to a short story, or even just a list of facts.

3.2. Language Processing and Analysis Module

This is the brilliant brain of our NLP system. It takes the incoming content and thoroughly breaks it down:

- **Language Detection:** First, it figures out what language the content is in.
- **Tokenization and POS Tagging:** Then, it's like meticulously dissecting sentences, breaking them into individual words and tagging each word with its grammatical role (like "noun," "verb," etc.).
- **Named Entity Recognition (NER):** It's excellent at spotting and categorizing important things like names of people, places, specific numbers, and more (Finkel et al., 2005).
- **Dependency Parsing:** This step helps it understand how words relate to each other in a sentence, mapping out the grammatical connections.
- **Semantic Role Labeling (SRL):** Finally, it identifies who did what to whom, where, and when, by pinpointing the actions and their participants in a sentence.

3.3. Multilingual Adaptation Layer

This crucial layer enables Prashnadyuti's multilingual capabilities. It could employ:

- **Pre-trained Multilingual Models:** We can use advanced models (like mBERT or XLM-R) that have already learned from huge amounts of text in many languages. This allows them to apply what they've learned to new languages, even with little or no specific training for that language (Devlin et al., 2019).
- **Neural Machine Translation (NMT):** For languages where direct QG models are sparse, translating the original text into a well-supported language (like English) to generate questions there, and then translating those questions and answers back into the original language (Bahdanau et al., 2015).
- **Language-Specific Pipelines:** For high-demand languages, dedicated, fine-tuned NLP tools and question-generation models might be used to ensure the questions are as accurate and natural-sounding as possible.

3.4. Question Generation (QG) Module

This is where the actual questions are born! This module uses clever techniques to craft questions from the processed content:

- **Rule-based QG:** It can follow specific linguistic rules to turn statements into questions. For example, it can easily create questions about numbers, dates, or other clear facts (Heilman & Smith, 2010).
- **Neural QG (Seq2Seq Models):** For more complex or nuanced questions, it employs advanced AI models (like Transformers) that have been trained on massive datasets of questions and answers. These models can generate questions that truly fit the context and lead to the correct answer (Sun & Li, 2020).
- **Fact-based Question Generation:** This component is specifically designed to extract straightforward facts and transform them into classic "Who," "What," "When," "Where," and "How many" questions.

3.5. Answer and Distractor Generation Module

Once a question is crafted, this module makes sure our quizzes have good answers and tricky, yet fair, incorrect options (what we call "distractors"):

- **Identify Correct Answer:** It precisely pinpoints the right answer directly from the original text.
- **Generate Distractors:** This is where it gets creative! It creates believable wrong answers using several strategies:
  - **Semantic Similarity:** Finding words or phrases that are similar in meaning to the correct answer but are still incorrect.
  - **Rule-based Perturbation:** Slightly changing numbers, dates, or names from the text to create misleading options (Labutov & Lipson, 2013).
  - **Negative Sampling:** Picking other related but incorrect entities from the text to serve as tempting wrong choices.

3.6. Quiz Formatting and Delivery Module

Finally, this module takes all the carefully generated questions, correct answers, and distractors and puts them into a beautiful, easy-to-use quiz format. Our interactive MCQ webpage is a perfect example of what this module can produce—it allows you to jump right in, answer questions, and instantly see how you did. Plus, with smart integrations like Firebase, we can track scores in real-time and even create leaderboards, making the quiz experience more engaging and community-oriented!

4. Application and Demonstration

Our MCQ Quiz Webpage, even in its simplified form, truly shows off what Prashnadyuti can do. Imagine this system at work:

- **□. □□□□□□ □□□□□ □□□□□□□□□ □□□□□□□□□□□ (The First Lesson: Prithu Vainya, the Original Farmer):** Prashnadyuti would analyze the Sanskrit text, identify key characters, events, and concepts related to the story of Prithu Vainya. It would then generate comprehension questions, vocabulary questions based on difficult words, and possibly questions about the moral or historical context of the lesson, ensuring that options are grammatically correct and culturally relevant in Sanskrit.
- **The "Golden Feather" Story:** Prashnadyuti would effortlessly pick out numbers (like "3 diamonds," "21 pearls," "99 gold coins") and generate questions such as, "How many diamonds did the little man offer for the golden feather?" It could also extract key vocabulary ("walk") and important plot points ("Queen found a feather while taking a walk") to craft comprehension questions.
- **The "Cuckoo" Poem:** The system would analyze all the time references ("April," "May," "June," "July," "August") and the cuckoo's actions ("open its bill," "sings all day," "change its tune," "fly away," "go away") to create questions about the bird's yearly journey and the poem's structure.

- **Number Work Problems:** For tricky math questions (like "expanded form," "place value," or "addition/subtraction"), Prashnadyuti would understand the math, perform the calculations, and then generate answer choices that include common mistakes, making them good distractors.
- **Roman Numerals:** The system would apply the specific rules of Roman numerals to not only generate questions but also to ensure the correct answer is perfect and the incorrect ones are believable, often by mimicking common errors (like `IIII` instead of `IV`).

The amazing thing about the multilingual aspect is that the same original content could be processed to create quizzes in English, Hindi, Sanskrit, or any other language we support, making learning truly accessible to a diverse student population.

5. Challenges and Future Work

Building a system like Prashnadyuti comes with its own set of exciting challenges:

- **Getting the Meaning Right:** We need to ensure that the questions we generate truly capture the essence of the original text and that the answers are absolutely correct.
- **Crafting Great Distractors:** Creating incorrect answer choices that are genuinely plausible yet definitively wrong is an art, and it's crucial for effective learning (Labutov & Lipson, 2013).
- **Navigating Language Nuances:** We need to be super careful to maintain natural phrasing, cultural relevance, and linguistic correctness across all supported languages.
- **Adapting to Any Topic:** The system needs to be smart enough to handle different subjects (like history, science, or literature) and adjust to various learning levels, from elementary school to university.
- **Being Fair and Unbiased:** It's vital to ensure our question generation doesn't introduce any biases and remains fair to all learners, regardless of their background.

Our future work will focus on:

- **Even Smarter AI Models:** We're excited to integrate more advanced large language models to make our questions and distractors even higher quality (Sun & Li, 2020).
- **Listening to Our Users:** We plan to build in ways for users to tell us what they think of the questions, so the system can constantly learn and improve.
- **Personalized Learning Journeys:** We want to develop the ability to generate questions that are perfectly tailored to each learner's skill level and how they prefer to learn.
- **More Question Types:** Beyond MCQs, we're looking to create fill-in-the-blanks, true/false, short answer, and even analytical questions.
- **Super Strong Multilingual Capabilities:** We'll continue to dig deep into how AI can learn across languages, aiming to generate questions that sound perfectly native, even for languages with fewer existing resources (Conneau et al., 2018).

6. Conclusion

Prashnadyuti offers a thrilling glimpse into the future of automated, multilingual question and answer generation, promising to transform how we create educational content. By bringing together robust NLP techniques, cutting-edge machine learning, and a smart, scalable design, this system can drastically lighten the load of manual question writing, provide a treasure trove of varied practice materials, and ultimately open doors to more accessible and personalized learning experiences around the globe. While there are certainly mountains to climb, the foundational ideas and the incredible pace of technological progress mean that ***Prashnadyuti*** is poised to become an indispensable tool in the world of education.

References

Bahdanau, D., Cho, K., & Bengio, Y. (2015). Neural Machine Translation by Jointly Learning to Align and Translate. *Proceedings of the International Conference on Learning Representations (ICLR)*.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., ... & Amodei, D. (2020). Language Models are Few-Shot Learners. *Advances in Neural Information Processing Systems, 33*.

Conneau, A., Kiela, D., Schwenk, H., Durme, B. V., & Lample, G. (2018). XNLI: Evaluating Cross-lingual Natural Language Inference. *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*.

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*.

Du, X., & Cardie, C. (2017). Learning to Ask: Neural Question Generation for Reading Comprehension. *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.

Finkel, J. R., Grenager, P., & Manning, C. (2005). Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling. *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*.

Heilman, M., & Smith, N. A. (2010). Towards a Shared Evaluation of Automatically Generated Questions. *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*.

Jurafsky, D., & Martin, J. H. (2009). *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Prentice Hall.

Labutov, I., & Lipson, H. (2013). Enhancing Peer Grading with Automated Distractor Generation. *Proceedings of the AAAI Conference on Artificial Intelligence*, *27*(1), 522-528.

Rajpurkar, P., Jia, R., & Liang, P. (2018). Know What You Don't Know: Unanswerable Questions for SQuAD 2.0. *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*.

Sun, S., & Li, R. (2020). QG-Net: A Neural Question Generation Model for Context-Aware Semantic Question Generation. *Proceedings of the AAAI Conference on Artificial Intelligence, 34*(05), 9037-9044.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention Is All You Need. *Advances in Neural Information Processing Systems, 30*.