SVNIT_CSE: Building a question answering system for Hindi using word-embedding

Ankur Jariwala[1,†], Siba Sankar Sahu[1,†]

[1]Department of Computer Science and Engineering, Sardar Vallabhbhai National Institute of Technology, Surat, India

Abstract

Question answering (QA) is an important application in the text analysis domain that enables machines to provide precise and factual answers. The rich morphology and diverse syntactic structures within Indian languages present both challenges and opportunities for the development of effective QA systems. As part of the FIRE 2025 VATIKA shared task, our SVNIT_CSE team explored different word embedding models for the Hindi QA system. From the evaluation results, we found that FastText embeddings with cosine similarity approach outperform other methods and provide a BLEU score and F_1 score of 89.5 and 0.9180 in test data I, respectively. Similarly, it provides the BLEU score and the F_1 score of 43.7 and 0.433 in test data II, respectively. The word embedding model contributes to the building of scalable, robust, and inclusive QA systems that can support access to information for millions of Indian language speakers.

Keywords

Low resource languages, Natural language processing, Question answering system, Word embeddings

1. Introduction

Question answering (QA) is an application of Information retrieval (IR) and Natural language processing (NLP) that answer user generated natural language queries. In general, the QA system is divided into two types based on the source of knowledge. The first type is extractive or closed domain; it uses span extraction from the available context to answer questions. The second type is generative, or open domain, which utilizes multiple large-scale data sources, such as the Web, documents, and journals, to answer questions. As people want quick and relevant information, so the application of QA system in different fields such as healthcare, education, business, and regular life through digital assistants.

In traditional search engine, the user gets a list of documents, websites, or references based on simple keyword matching; however, a QA system provides concise, accurate, and contextually relevant answers by understanding the user's query. The most important features of a QA system is their ability to understand natural language queries and provide relevant information from structured or unstructured sources. The QA systems and Generative AI based conversational tools are a lot different. Large language models like ChatGPT would operate in a much broader scope and ability to keep track of context across multiple turns and perform in a more natural, open-ended manner. However, the QA system provides answers in a specific domain.

Advanced search engines, virtual assistants, and a conversational AI system are built for high-resource languages such as English and European languages [1] [2]. However, there is less availability of such AI system in low- resource languages. Developing an AI system for low resource languages closes the digital divide and makes it easier for people to access digital services, healthcare, and education. The development of QA systems for low-resource languages is of immense importance in making information access more inclusive and equitable. In this study, we explore a QA system for the low-resource Indian language.

FIRE'25: Forum for Information Retrieval Evaluation, December 17–20, 2025, India

†These authors contributed equally.

p24ds008@coed.svnit.ac.in (A. Jariwala); sibasankar@coed.svnit.ac.in (S. S. Sahu)

0009-0008-1688-9699 (A. Jariwala); 0000-0001-9769-9206 (S. S. Sahu)

published 2026-02-24

1.1. Task Description

FIRE [1] (Forum for Information Retrieval Evaluation) organized a shared task on the QA system for Varanasi Tourism (VATIKA [2]). The VATIKA [3] data set comprises ten different data from the tourism domain. Domains are included, such as Ganga aarti, cruise, food court, public toilet, kund, museum, general, ashram, temple, and travel. Each domain includes detailed paragraph-level Hindi contexts followed by multiple QA pairs. The data set is divided into four main parts i.e; train, validation, test data I, and test data II. QA pairs of different training data comprise the unique ID, question, and answer. An example of a Hindi QA pair training data and test data is shown in Fig. 1 and Fig. 2. In test data II, the user generates a predicted answer for a particular query. The statistic of the VATIKA data set is shown in Table 1.


```
{
  "domains": [
    {
      "domain": "kund",
      "contexts": [
        {
          "context": "भागीरथ कुंड पं. दीन दयाल उपाध्याय रेलवे स्टेशन से 14.1 किलोमीटर दूर है। स्टेशन से कुंड तक पहुँचने के लिए टैक्सी, कैब, या बस सेवाओं का उपयोग किया जा सकता है। यह स्टेशन पूर्व में मुगलसराय के नाम से जाना जाता था और भारत के प्रमुख रेल जंक्शनों में से एक है। यहाँ से भागीरथ कुंड तक की यात्रा वाराणसी की ऐतिहासिक गलियों और घाटों के दृश्य प्रदान करती है। इस यात्रा में भक्तों को वाराणसी की सांस्कृतिक विरासत का अनुभव मिलता है, जो इस धार्मिक स्थल के महत्व को और भी बढ़ा देता है।",
          "qas": [
            {
              "id": "kund_636",
              "question": "भागीरथ कुंड पं. दीन दयाल उपाध्याय रेलवे स्टेशन से कितना किलोमीटर दूर है? ",
              "answer": "भागीरथ कुंड पं. दीन दयाल उपाध्याय रेलवे स्टेशन से 14.1 किलोमीटर दूर है।"
            },
            {
              "id": "kund_637",
              "question": "भागीरथ कुंड पं. दीन दयाल उपाध्याय रेलवे स्टेशन से कैसे पहुँच सकते है।",
              "answer": "भागीरथ कुंड तक पहुँचने के लिए पं. दीन दयाल उपाध्याय रेलवे स्टेशन से टैक्सी, कैब, या बस सेवाओं का उपयोग किया जा सकता है।"
            }
          ]
        }
      ]
    }
  ]
}
```


Figure 1: Example of train, validation, test data I in VATIKA data set

[1] https://fire.irsi.org.in/

[2] https://sites.google.com/view/vatika-2025/

```
{
  "domains": [
    {
      "domain": "kund",
      "contexts": [
        {
          "context": "आदिकेशव कुंड, जो वाराणसी के प्राचीन राजघाट के निकट स्थित है, वरुणा एवं गंगा नदियों के पावन संगम स्थल पर स्थित एक अत्यंत धार्मिक और सांस्कृतिक दृष्टि से महत्त्वपूर्ण स्थान है। इसे भगवान विष्णु को समर्पित एक दिव्य स्थल के रूप में पूज्य माना जाता है। भारतीय संस्कृति एवं धर्मशास्त्रों में गंगा–वरुणा के इस संगम का अत्यधिक आध्यात्मिक एवं सांस्कृतिक महत्व है, जिसे स्नानादि कर्मों के माध्यम से पापों से मुक्ति और पुण्य प्राप्ति का अभिन्न स्रोत माना जाता है। श्रद्धालु इस कुंड में स्नान कर आत्मशुद्धि की अभिलाषा रखते हैं तथा इसे वाराणसी की समृद्ध ऐतिहासिक एवं धार्मिक विरासत का अनिवार्य अंग मानते हैं। यह स्थल वाराणसी की ऐतिहासिक और धार्मिक धरोहर का हिस्सा है।",
          "qas": [
            {
              "id": "kund_001",
              "question": "आदिकेशव कुंड किन नदियों के संगम पर स्थित है और इसका धार्मिक महत्व क्या है? ",
              "answer": ""
            },
            {
              "id": "kund_002",
              "question": "आदिकेशव कुंड का संबंध किस देवता से है और श्रद्धालु यहाँ क्यों आते हैं? ",
              "answer": ""
            },
            {
              "id": "kund_003",
              "question": "आदिकेशव कुंड को वाराणसी की धार्मिक विरासत में कैसे देखा जाता है? ",
              "answer": ""
            }
          ]
        }
      ]
    }
  ]
}
```


Figure 2: Example of test data II in VATIKA data set

Table 1
VATIKA Dataset

Split	Contexts	QA Pairs
Train	5244	13092
Validation	1134	2798
Test Data-I	1143	2902
Test Data-II	430	1196

The paper is organized as follows. Section 2 presents previous research conducted in the QA domain. The preprocessing step and the model architecture are presented in Section 3. The experimental results and their analysis are described in Section 4. Finally, we conclude with the direction of future work in Section 5.

2. Related Work

Early research work on QA systems relied heavily on human effort, both in data curation and annotation. Researchers often used Wikipedia articles, newswire text, and encyclopedias as sources of factual knowledge because there were not many large-scale machine-readable datasets available. In recent years, several methods have been explored, such as machine learning, deep learning, and transformer-based models to deal with the QA system. In this section, we look at some existing studies on QA systems.

QA systems for English and other high-resource languages have come a long way [4] [5]. However, a few research has been conducted on low-resource Indian languages. Nanda et al. [6] implemented a machine learning approach for the Hindi QA system. The approach includes accessing natural language queries, feature extraction, and classification. They trained a Naive Bayes classifier to identify the correct label class for a given query. They evaluated the performance on two test data and achieved an accuracy of 92% and 88%. Hermjakob [7] combined the Penn treebank training corpus with the question treebank for the classification of questions. The experimental results show that by adding question sentences for training, the accuracy increases from 65.5% to 97.3%. Zhang and Lee [8] evaluated the five machine learning algorithms: nearest neighbors, naïve bayes, decision tree, sparse network of windows, and support vector machine. They trained the learning algorithms on different length of datasets and tested on the TREC [3] dataset. They found that the SVM algorithm outperformed the four other methods in the question classification.

Abdel-Nabi et al. [9] report a survey on different deep learning models to build a QA system. They found that different deep learning models such as the convolutional neural network, recurrent neural network, attention-based, hybrid models, graph-based models, generative model, and reinforcement learning-based models were used for different QA systems. They evaluated the effectiveness of the QA system in a wide range of evaluation metric. Tan et al. [10] used the BiLSTM model to generate the embeddings of the questions and answers. They used two data sets, i.e. TREC-QA [4] and InsuranceQA [5] to evaluate an QA system. The QA-LSTM/CNN with attention mechanism provides the best performance on TREC-QA. Similarly, QA-LSTM with attention offers the best performance in InsuranceQA. Tay et al. [11] proposed holographic dual LSTM (HD-LSTM) for QA system. They found that extensive feature engineering was not required to build a deep learning-based QA system. They noticed that the MAP score for HD-LSTM is 0.7042 which outperformed the baseline LSTM (1 layer) with a MAP score of 0.6280. Similarly, the HD-LSTM MRR score is 0.7733 outperformed the baseline LSTM (1 layer) with a MRR score of 0.6960. Peng and Liu [12] presented an attention-based convolutional neural network model for the QA system. The experimental results show that the attention-based convolutional neural network performs better than CNN and the LSTM model.

Several benchmark data sets have been created to test cross-lingual abilities such as XQuAD (Artetxe et al. [13]), MLQA (Lewis et al. [14]), and TyDi QA (Clark et al. [15]). These corpora use a translation-based approach or parallel corpora to cover many languages. These resources enable a systematic evaluation of cross-lingual transfer, where models are trained in high-resource languages and tested in low-resource settings. The data set also helps advance research on multilingual QA. In Indian languages, there is less availability of annotated QA datasets on the Web. Hence, the researcher explored zero-shot transfer [14], few-shot fine-tuning [16], and synthetic data generation [17] to build their own customized datasets. These strategies are helpful, but the more complex morphology of Indian languages often limits their effectiveness.

To address these limitations, several Indian language datasets have been developed. IndicSQuAD [18], dataset is developed by translating the English SQuAD dataset into ten major Indian languages. The resource is fine-tuned with both monolingual BERT and multilingual models such as MuRIL-BERT (Khanuja et al. [19]). They found that language-specific BERT outperform MurilBERT in different Indian languages. IndicQA is another tool that presents a standard for both extractive and abstractive QA in

[3] https://huggingface.co/datasets/CogComp/trec
[4] https://huggingface.co/datasets/lucadiliello/trecqa
[5] https://huggingface.co/datasets/deccan-ai/insuranceQA-v2

different Indian languages [20]. They explored different LLM for QA. They explored two inferences, that is, translate test and direct test in different Indian languages. In the translate test, the input is translated into English, and the output is translated into Indian language. In the direct test, both the input and the output are in the native language. They found that the translate test provides better performance than the direct test in all Indian languages.

IndicBERT (Doddapaneni et al. [21]) is a lightweight transformer model built on the ALBERT architecture and trained on IndicCorp, a large collection of monolingual Indian languages. IndicBERT supports zero-shot transfer and cross-lingual generalization, allowing a model trained in one Indian language to transfer knowledge to others. Compared to multilingual models such as mBERT [22] or XLM-RoBERTa [23], IndicBERT requires fewer parameters while maintaining competitive performance, making it more suitable for languages with low resources. Doddapaneni et al. [21] show that the combination of IndicBERT and Samanantar provides a better avg F_1 score than MuRIL and mBERT in different Indian languages. Samanatar [24] is the largest publicly available parallel corpora collection for eleven Indian languages. Together, these datasets and models are the building blocks for furthering QA research in Indian languages.

3. Proposed Methodology

We explore the QA system in three steps: data pre-processing, model design, and evaluation. In the preprocessing step, the data set is tokenized, segmented, and presented in a structured format. The QA model is designed using traditional similarity-based methods with modern embedding-based approaches. Finally, we evaluate the QA system using standard evaluation metrics.

3.1. Data Preprocessing

The data set is presented in SQuAD-like [25] JSON structure that contains context, question, and answer. We implemented different preprocessing steps and presented the data in a structured tabular format. Then, each QA pair is presented in the following way.

Q = question, C= { c1,c2,...,cn }, Atext = ground truth answer text. where C is the set of candidate sentences in the context. The goal is to identify the sentence in context C that best answers the question Q. For every method, the question and context sentences are converted into a vector space, and the similarity is measured.

3.2. Model Architecture

Word embeddings are compact, low-dimensional vector representations of words that contain syntactic and semantic characteristics of words within a continuous space. Embeddings place words with similar meanings nearer to each other in the vector space. This feature enables them to be very efficient for NLP activities such as named entity recognition [26] and sentiment analysis [27]. In this study, we explore different word embedding models to capture semantic similarity between question and answer and develop an efficient QA system.

3.2.1. FastText embeddings with cosine similarity approach

We use pre-trained FastText [28] embeddings [6] to represent words in a dense vector space. We tokenize the input text, and the embedding method presents each word to a 300-dimensional embedding vector. The mean embeddings of all words in the sentence to obtain sentence-level representations for both the context and the question. The averaging produces fixed-size sentence vectors that capture the semantic meaning of the text. For each question–answer pair, we represent the context and the question as separate vectors and compute the cosine similarity between the question vector and the sentence vectors of the context. The sentence with the highest cosine similarity score is selected as the predicted

[6] https://fasttext.cc/docs/en/crawl-vectors.html

answer. The approach assumes that semantically similar sentences, based on the embedding space, are more likely to contain the correct answer span.

3.2.2. FastText embeddings with machine learning approach

We explore pre-trained FastText embeddings to represent text in a continuous vector space. Individually, word is mapped to a 300-dimensional embedding. The sentence representations for both context and question are received by averaging the embeddings of their constituent words. For every question–answer pair, we calculate a context vector and a question vector, then concatenated to construct the final feature representation. In the dataset, the annotated answer positions are used to make labels that match the training samples. These concatenated vectors are used to train a ridge regression model. The system learns to connect the input representation to the correct answer spans. The performance of the model is evaluated using the predicted and actual answers.

3.2.3. Word2Vec Embeddings

We investigate pre-trained Word2Vec [29] embeddings to obtain distributed representations of words. A fixed dimension dense vector is used to represent each token in the dataset, regardless of whether it originates from the context or the question. Word2Vec captures semantic similarity by putting words with similar meanings closer together in the embedding space. This is especially beneficial for dealing with different Hindi vocabulary. The system uses these embeddings to turn both context passages and questions into machine-readable form that keeps the meaning of the words. To make sentence-level representations, we determine the average of the Word2Vec embeddings of all the tokens in a certain text span. This gives us vectors of the same size for both the question and each sentence in the context. The idea is that the aggregated embedding captures the main meaning of the text span, allowing to compare the question and context in the same vector space. Then, we use cosine similarity to check similarity in the sentence and question vector in the embedding space. We used the following parameter to experiment with the Word2Vec model which is presented below in Table 2.

Table 2
Parameters of Word2Vec model

Parameter Name	Value
vector_size	100
window	5
min_count	1
workers	4

4. Results and Analysis

In the QA system, we explore different embedding-based approaches to extract the answers from the context. To evaluate the effectiveness of the system, we use evaluation metrics such as the BLEU score, the ROUGE score, and the F_1 score. The evaluation score of different embedding models is presented in Tables 3-8. Moreover, the effectiveness of different word embedding models is presented graphically in Figures 3a-5b. From the evaluation results, we found that the FastText method with cosine similarity (CS) outperforms other methods in both Test data I and II in terms of BLEU and F_1 score. A better BLEU and F_1 scores says that FastText embeddings are better at getting exact matches and subword-level accuracy. Word2Vec model provides the best ROUGE scores in test data I as shown in Table 5 whereas Fasttext with machine learning offers the best ROUGE scores in test data II as shown in Table 6. A better ROUGE score indicates that FastText with the ML model captures broader contextual and structural similarity. In Table 4, FastText with cosine similarity (CS), BLEU scores start relatively high at the unigram level but decrease with higher n-grams, reaching very low at BLEU-4. The results demonstrate

that the ground truth answer is present in the system-generated output but not sequentially. From the analysis of the results, we found that FastText with CS provides optimal performance on different evaluation metric and provides competitive performance on shared task leaderboard. The evaluated model is most suitable for the Hindi QA system.

Table 3
BLEU scores using different methods on Test Data I

Method	BLEU-1	BLEU-2	BLEU-3	BLEU-4
FastText + Cosine Similarity	**89.50**	**88.53**	**88.10**	**87.70**
Word2Vec	83.23	82.23	81.56	80.71
FastText + Machine Learning	67.47	66.65	66.13	65.36

Table 4
BLEU scores using different methods on Test Data II

Method	BLEU-1	BLEU-2	BLEU-3	BLEU-4
FastText + Cosine Similarity	**43.7**	**22.9**	**14.8**	**10.8**
Word2Vec	41.3	20.9	14.3	10.6
FastText + Machine Learning	39.3	17.5	10.6	7.6

Table 5
ROUGE scores using different methods on Test Data I

Method	ROUGE-1	ROUGE-2	ROUGE-L
Word2Vec	**0.4715**	**0.2956**	**0.4715**
FastText + Cosine Similarity	0.4604	0.2845	0.4604
FastText + Machine Learning	0.3702	0.2182	0.3702

Table 6
ROUGE Scores using different methods on Test Data II

Method	ROUGE-1	ROUGE-2	ROUGE-L
FastText + Machine Learning	**0.0790**	**0.0452**	**0.0790**
FastText + Cosine Similarity	0.0577	0.0260	0.0577
Word2Vec	0.0552	0.0226	0.0552

Table 7
QA-F_1 score using different methods on Test Data I

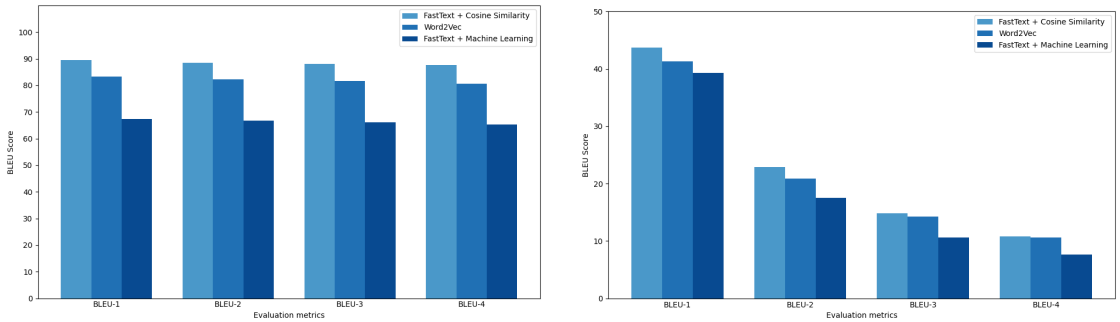
Method	QA-F_1
FastText + Cosine Similarity	**0.9180**
Word2Vec	0.8330
FastText + Machine Learning	0.6711

5. Conclusion

The question-and-answer is an important downstream task in the text analysis domain. In this study, we explore various embedding-based methodologies for developing a QA system in Hindi. From the

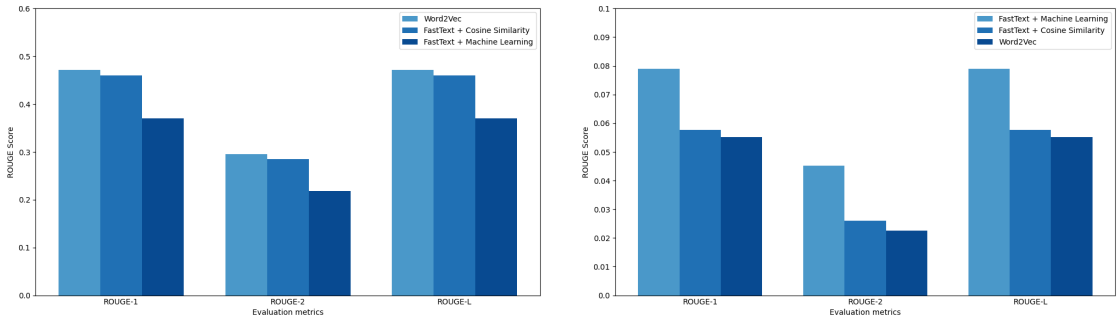
Table 8
QA-F_1 score using different methods on Test Data II

Method	**QA-F_1**
FastText + Cosine Similarity	**0.433**
Word2Vec	0.394
FastText + Machine Learning	0.280



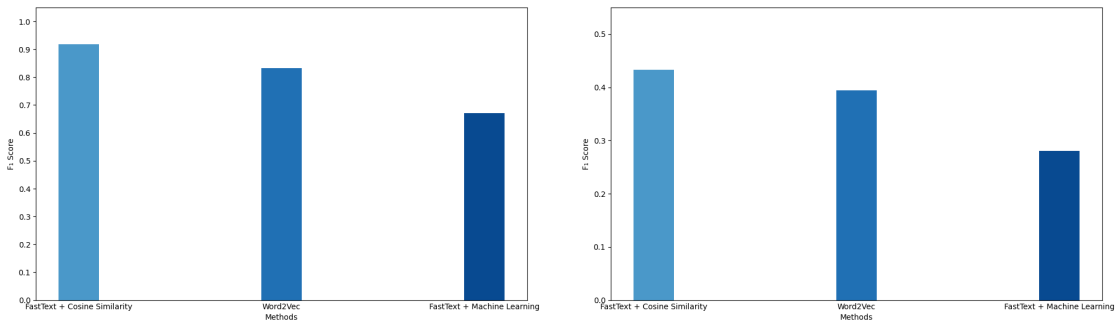
(a) (b)

Figure 3: (a) QA Performance BLEU score on Test Data I (b) QA Performance BLEU score on Test Data II



(a) (b)

Figure 4: (a) QA Performance ROUGE score on Test Data I (b) QA Performance ROUGE score on Test Data II



(a) (b)

Figure 5: (a) QA Performance QA-F1 score on Test Data I (b) QA Performance F1 score on Test Data II

evaluation results, we found that FastText with cosine similarity outperformed other methods, achieving the highest BLEU and QA-F_1 scores on different test data. In general, the results show that simpler similarity-based methods provide better performance for Hindi QA than the traditional machine learning method. In the future, we can explore transformer-based models to build the Hindi QA system and improve the robustness in real-world applications.

Declaration on Generative AI

During the preparation of this work, the author(s) used ChatGPT, QuillBot in order to: Spelling Check, Paraphrase and reword. After using this tool/service, the author(s) reviewed and edited the content as needed and take(s) full responsibility for the publication's content.

References

[1] S. Liaw, H. Huang, An investigation of user attitudes toward search engines as an information retrieval tool, Comput. Hum. Behav. 19 (2003) 751–765.

[2] C. V. Gysel, Modeling spoken information queries for virtual assistants: Open problems, challenges and opportunities, in: H. Chen, W. E. Duh, H. Huang, M. P. Kato, J. Mothe, B. Poblete (Eds.), Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2023, Taipei, Taiwan, July 23-27, 2023, ACM, 2023, pp. 3335–3338.

[3] P. Gatla, Anushka, N. Kanwar, G. Sahoo, R. K. Mundotiya, Tourism question answer system in indian language using domain-adapted foundation models, arXiv preprint (2025).

[4] R. F. Simmons, Answering english questions by computer: a survey, Commun. ACM 8 (1965) 53–70.

[5] O. Kolomiyets, M. Moens, A survey on question answering technology from an information retrieval perspective, Inf. Sci. 181 (2011) 5412–5434.

[6] G. Nanda, M. Dua, K. Singla, A hindi question answering system using machine learning approach, in: 2016 International Conference on Computational Techniques in Information and Communication Technologies (ICCTICT), 2016, pp. 311–314.

[7] U. Hermjakob, Parsing and question classification for question answering, in: Proceedings of the ACL 2001 Workshop on Open-Domain Question Answering, 2001.

[8] D. Zhang, W. S. Lee, Question classification using support vector machines, in: Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval, SIGIR '03, Association for Computing Machinery, New York, NY, USA, 2003, p. 26–32.

[9] H. Abdel-Nabi, A. Awajan, M. Z. Ali, Deep learning-based question answering: a survey, Knowledge and Information Systems 65 (2023) 1399–1485.

[10] M. Tan, C. dos Santos, B. Xiang, B. Zhou, Lstm-based deep learning models for non-factoid answer selection, 2016.

[11] Y. Tay, M. C. Phan, L. A. Tuan, S. C. Hui, Learning to rank question answer pairs with holographic dual lstm architecture, in: Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '17, ACM, 2017, p. 695–704.

[12] Y. Peng, B. Liu, Attention-based neural network for short-text question answering, in: Proceedings of the 2018 2nd International Conference on Deep Learning Technologies, ICDLT '18, Association for Computing Machinery, New York, NY, USA, 2018, p. 21–26.

[13] M. Artetxe, S. Ruder, D. Yogatama, On the cross-lingual transferability of monolingual representations, in: D. Jurafsky, J. Chai, N. Schluter, J. R. Tetreault (Eds.), Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020, Association for Computational Linguistics, 2020, pp. 4623–4637.

[14] P. Lewis, B. Oguz, R. Rinott, S. Riedel, H. Schwenk, MLQA: evaluating cross-lingual extractive question answering, in: D. Jurafsky, J. Chai, N. Schluter, J. R. Tetreault (Eds.), Proceedings of the

58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020, Association for Computational Linguistics, 2020, pp. 7315–7330.

[15] J. H. Clark, J. Palomaki, V. Nikolaev, E. Choi, D. Garrette, M. Collins, T. Kwiatkowski, Tydi QA: A benchmark for information-seeking question answering in typologically diverse languages, Trans. Assoc. Comput. Linguistics 8 (2020) 454–470.

[16] J. Hu, S. Ruder, A. Siddhant, G. Neubig, O. Firat, M. Johnson, XTREME: A massively multilingual multi-task benchmark for evaluating cross-lingual generalisation, in: Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event, volume 119 of *Proceedings of Machine Learning Research*, PMLR, 2020, pp. 4411–4421.

[17] R. Puri, R. Spring, M. Shoeybi, M. Patwary, B. Catanzaro, Training question answering models from synthetic data, in: B. Webber, T. Cohn, Y. He, Y. Liu (Eds.), Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020, Association for Computational Linguistics, 2020, pp. 5811–5826.

[18] S. Endait, R. Ghatage, A. Kulkarni, R. Patil, R. Joshi, Indicsquad: A comprehensive multilingual question answering dataset for indic languages, CoRR abs/2505.03688 (2025).

[19] S. Khanuja, D. Bansal, S. Mehtani, S. Khosla, A. Dey, B. Gopalan, D. K. Margam, P. Aggarwal, R. T. Nagipogu, S. Dave, S. Gupta, S. C. B. Gali, V. Subramanian, P. P. Talukdar, Muril: Multilingual representations for indian languages, CoRR abs/2103.10730 (2021).

[20] A. K. Singh, V. Kumar, R. Murthy, J. Sen, A. R. Mittal, G. Ramakrishnan, INDIC QA BENCHMARK: A multilingual benchmark to evaluate question answering capability of llms for indic languages, in: L. Chiruzzo, A. Ritter, L. Wang (Eds.), Findings of the Association for Computational Linguistics: NAACL 2025, Albuquerque, New Mexico, USA, April 29 - May 4, 2025, Association for Computational Linguistics, 2025, pp. 2607–2626.

[21] S. Doddapaneni, R. Aralikatte, G. Ramesh, S. Goyal, M. M. Khapra, A. Kunchukuttan, P. Kumar, Towards leaving no indic language behind: Building monolingual corpora, benchmark and models for indic languages, in: A. Rogers, J. L. Boyd-Graber, N. Okazaki (Eds.), Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023, Association for Computational Linguistics, 2023, pp. 12402–12426.

[22] S. Wu, M. Dredze, Beto, bentz, becas: The surprising cross-lingual effectiveness of BERT, in: K. Inui, J. Jiang, V. Ng, X. Wan (Eds.), Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019, Association for Computational Linguistics, 2019, pp. 833–844.

[23] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, E. Grave, M. Ott, L. Zettlemoyer, V. Stoyanov, Unsupervised cross-lingual representation learning at scale, in: D. Jurafsky, J. Chai, N. Schluter, J. R. Tetreault (Eds.), Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020, Association for Computational Linguistics, 2020, pp. 8440–8451.

[24] G. Ramesh, S. Doddapaneni, A. Bheemaraj, M. Jobanputra, R. AK, A. Sharma, S. Sahoo, H. Diddee, M. J, D. Kakwani, N. Kumar, A. Pradeep, S. Nagaraj, D. Kumar, V. Raghavan, A. Kunchukuttan, P. Kumar, M. S. Khapra, Samanantar: The largest publicly available parallel corpora collection for 11 indic languages, Trans. Assoc. Comput. Linguistics 10 (2022) 145–162.

[25] P. Rajpurkar, J. Zhang, K. Lopyrev, P. Liang, Squad: 100, 000+ questions for machine comprehension of text, in: J. Su, X. Carreras, K. Duh (Eds.), Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016, The Association for Computational Linguistics, 2016, pp. 2383–2392.

[26] A. Das, D. Ganguly, U. Garain, Named entity recognition with word embeddings and wikipedia categories for a low-resource language, ACM Trans. Asian Low Resour. Lang. Inf. Process. 16 (2017) 18:1–18:19.

[27] T. P. Adewumi, F. Liwicki, M. Liwicki, Word2vec: Optimal hyper-parameters and their impact on NLP downstream tasks, CoRR abs/2003.11645 (2020).

[28] P. Bojanowski, E. Grave, A. Joulin, T. Mikolov, Enriching word vectors with subword information, Trans. Assoc. Comput. Linguistics 5 (2017) 135–146.

[29] T. Mikolov, K. Chen, G. Corrado, J. Dean, Efficient estimation of word representations in vector space, in: Y. Bengio, Y. LeCun (Eds.), 1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings, 2013.