Hindi Chatbot for Supporting Maternal and Child Health Related Queries in Rural India

Ritwik Mishra[1] **, Simranjeet Singh**[2] **, Jasmeet Kaur**[1] **, Pushpendra Singh**[1]
and **Rajiv Ratn Shah**[1]
[1] Indraprastha Institute of Information Technology, Delhi
{ritwikm, jasmeetk, psingh, rajivratn}@iiitd.ac.in
[2] Netaji Subhas University of Technology, Delhi
simranjeets.ec18@nsut.ac.in

Abstract

In developing countries like India, doctors and healthcare professionals working in public health spend significant time answering health queries that are fact-based and repetitive. Therefore, we propose an automated way to answer maternal and child health-related queries. A database of Frequently Asked Questions (FAQs) and their corresponding answers generated by experts is curated from rural health workers and young mothers. We develop a Hindi chatbot that identifies k relevant Question and Answer (QnA) pairs from the database in response to a healthcare query (q) written in Devnagri script or Hindi-English (Hinglish) code-mixed script. The curated database covers 80% of all the queries that a user of our study is likely to ask. We experimented with (i) rule-based methods, (ii) sentence embeddings, and (iii) a paraphrasing classifier, to calculate the q-Q similarity. We observed that paraphrasing classifier gives the best result when trained first on an open-domain text and then on the healthcare domain. Our chatbot uses an ensemble of all three approaches. We observed that if a given q can be answered using the database, then our chatbot can provide at least one relevant QnA pair among its top three suggestions for up to 70% of the queries.

1 Introduction

With inequality in healthcare access across urban and rural parts of India, pregnant and postpartum women in rural areas suffer from low access to healthcare due to limited time with healthcare professionals, language barriers in doctor-patient communication, and societal barriers. In resource-constrained environments, digital support groups are a common platform to seek information about various maternal and child healthcare-related issues (Das and Sarkar, 2014; Kaur et al., 2019; Yadav et al., 2022). The moderators of such support groups are overburdened with enormous queries and find it challenging to provide answers timely. Moreover, group members often ask their health queries in regional languages such as *Hindi* or Hinglish[1]. Given the doctor-to-population ratio of 4.8 doctors per 10000 people in India (Potnuru et al., 2017), the scalability of such healthcare interventions involving doctors becomes challenging (Kaur et al., 2019). Thus, it presents an opportunity to extend informational support to pregnant and postpartum women through a chatbot that can answer their written queries in their local language.

Chatbots are used in various domains, from railways ticket reservations to food delivery[2]. Chatbots have taken up different roles in healthcare, such as psychotherapists, nurses, doctors, and medical consultants (Weizenbaum, 1966; Agrawal et al., 2017; Comendador et al., 2015). Chatbots have the potential to act as the first point of contact for women seeking answers for maternal and child healthcare-related queries, especially in resource-constrained environments (Yadav et al., 2019b). In this work, we explore the potential of a chatbot to provide accurate healthcare information by retrieving the best matching FAQs with their corresponding answers (Mittal et al., 2021).

We developed a chatbot that provides k most relevant FAQs with their corresponding answers (QnA pairs) in response to a healthcare query. The chatbot uses a curated database of QnA pairs in the Hindi language with answers vetted by healthcare professionals. Our chatbot can process user queries written in Latin script (native script for English) and Devanagari script (native script for Hindi). Figure 1 illustrates the overall architecture of the proposed chatbot. For evaluation, we obtained a set of healthcare queries from ASHA

[1] It is a colloquial term to describe a language written using the English script (Latin), but the grammar and vocabulary are borrowed from Hindi. It is also called Hindi-English code-mixed language. For example, 'नमस्ते '(*hello*) is written as '*namaste*'.

[2] https://www.chatbotguide.org/dominospizza-bot

Proceedings of the 5th Clinical Natural Language Processing Workshop, pages 69–77
July 14, 2023 ©2023 Association for Computational Linguistics

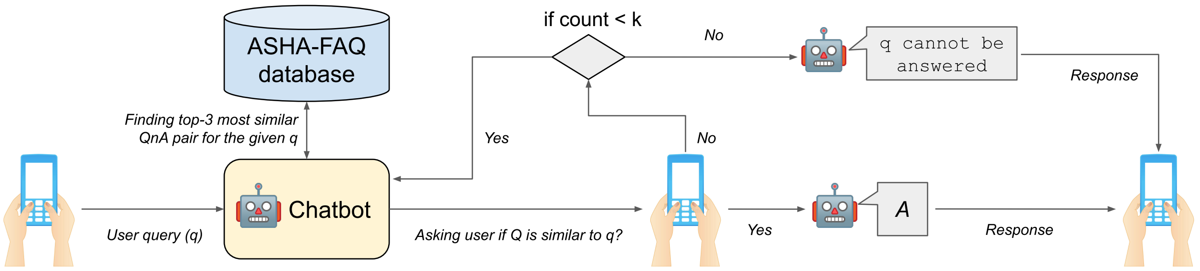

Figure 1: The architecture of the proposed chatbot is shown above. A user query (q) can be in the Devanagari or Latin script. The chatbot fetches top-k most similar Question-Answer (QnA) pairs from the ASHA-FAQ database and shows the user one question (Q) at a time.

workers[3]. In this paper, we discuss different algorithmic approaches to developing chatbots and the efficiency of these approaches in providing relevant QnA pairs. The three primary approaches used in this work are (i) the rule-based method, (ii) sentence embeddings, and (ii) paraphrasing classifiers. An ensemble model of all three primary approaches was found to be performing better than other methods. We release the source code of our chatbot to encourage future research in this direction [4].

2 Related Work

Earlier works on developing chatbots in healthcare using AI started with user query reformulation and using knowledge from search engines (Brill et al., 2002). They were made for the English language, and the same techniques could not be used for Hindi speakers due to the scarcity of resources. Kothari et al. (Kothari et al., 2009) aimed to develop a FAQ retrieval system for the unstructured English language written as a shorthand for SMS by the Indian population. It relied on character-level features to calculate the sentence similarity scores. Initial works on building a QnA system for the Hindi language were restricted to exploiting information from shallow speech features like POS tags (Sahu et al., 2012). In constructing an automatic question-answering system for English-Hindi code-switched language (also known as *Hinglish*), the word-level translation of code-switched queries to English queries was a common practice due to a lack of resources in the Hindi language (Raghavi et al., 2015; Sekine and Grishman, 2003). Such approaches fail to generalize because Hindi-to-English word-level translations are highly dependent on the position of the Hindi word in the sentence (Ray et al., 2018).

Previously cross-lingual word embeddings have been used to solve a healthcare QnA system in low-resource African languages(Daniel et al., 2019). It has been empirically shown that fine-tuned machine learning models using embeddings from pre-trained transformer-based encoders like BERT outperform many other traditional AI models on various tasks(González-Carvajal and Garrido-Merchán, 2020; Hao et al., 2019). Earlier works have shown the efficiency of BERT-based models in measuring sentence similarity for FAQ retrieval tasks(Bhagat et al., 2020; Sakata et al., 2019).

In this paper, we compared the performance of different approaches for measuring sentence similarity between Hindi sentences from the maternal healthcare domain. For a given user query (q), the most similar question (Q) and its corresponding answer (A) are fetched from the ASHA-FAQ database, which is described in the next section.

3 Data Description

We collected data from four prior studies by taking permission from the authors (Yadav et al., 2019a,b, 2021, 2017). The data consists of hundreds of pairs of questions and answers (in audio and text modality), as asked in the real world by community health workers and pregnant and postpartum women regarding maternal and child health issues. Health experts have provided the answers to these questions. The audio data was transcribed and annotated with the help of two healthcare professionals. Both annotators had a bachelor degree in medicine and surgery, a master's in public health, and experience working in maternal and child health. The two annotators manually transcribed each session in the Devanagari script.

[3] They are Accredited Social Health Activists (ASHA) employed by the Ministry of Health and Family Welfare, India. They are frontline health workers connecting the rural population with the state health system.

[4] github.com/ritwikmishra/asha-chatbot

In this work, annotations were performed using an online transliteration tool[5] and Audino (Grover et al., 2020). More than 18 hours of audio in the healthcare domain were transcribed to obtain 1150 question-answer (QnA) pairs. Subsequently, we received 217 maternal health question-answer pairs from Yadav et al. (Yadav et al., 2019b) and added them to our ASHA-FAQ database resulting in a total of 1365 unique questions and 1338 unique answers [6].

Due to the COVID-19 pandemic, AI model field testing was not feasible. Therefore, to evaluate the models on real-time data, a total of 336 new user queries (q) were collected from ASHA workers with the help of a non-governmental organization (NGO) partner. We requested ASHA workers to provide us with queries that they frequently encounter. With the help of public health professionals (with a master's degree in public health), the authors annotated these 336 queries with relevant questions from the ASHA-FAQ database. For each query q, authors identified completely and partially matching QnA pairs from the ASHA-FAQ database. Both types of matching (complete and partial) have been treated as relevant in this work. It has been found that, among 336 queries, 270 user queries had at least one relevant question in the database. Hence, the coverage of the ASHA-FAQ database is 80% in our experiment. The 270 questions, as mentioned above, will be treated as the *hold-out test set* to evaluate the performance of different FAQ-retrieval approaches used by our chatbot.

In order to train a deep-learning model to calculate the sentence similarity score between two Hindi sentences, we scraped Hindi news articles from the Inshorts website[7]. Each data point (d_i) in the scraped dataset (D) consisted of news article text (t_i), its headline (h_i), a summary of the text (t_i^s), and a paraphrased headline (h_i^p). We collected more than $17K$ data points in our dataset. For a negative (or not-paraphrased) headline of h_i, a random headline is chosen from the paraphrased headlines[8].

Our Inshorts dataset contains 35K Hindi sentences, from the news domain, classified into two classes (paraphrased or not-paraphrased), with equal representation of both classes. We are releasing the scraping scripts and hyperlinks to the news articles in the repository mentioned above. To the best of our knowledge, it is the most expansive dataset available for paraphrase detection in the Hindi language. Since the Inshorts dataset is from the open domain (news), we constructed a question paraphrase dataset in the healthcare domain. We manually paraphrased questions from the ASHA-FAQ database and treated them as positive examples of paraphrases. Random sentences were taken as negative examples. The dataset thus created is called the *AshaQs* dataset, and it contains about 1500 healthcare-related question pairs classified into two classes (paraphrased or not-paraphrased) in a balanced manner.

The performance of different FAQ retrieval models is compared using five information retrieval evaluation metrics, namely: Mean Average Precision (mAP), Mean Reciprocal Rank (MRR), Success Rate (SR), normalized Discounted Cumulative Gain (nDCG), and Precision at 3 (P@3) (Sakata et al., 2019). Success Rate is the simplest to understand because it represents the percentage of user queries for which at least one relevant suggestion was given in the top-k suggestions.

4 Methodology

Our work aims to take input as a user query (q) and produces an output as top-k most relevant QnA pairs from the ASHA-FAQ database. Therefore, the given task is modeled as a FAQ retrieval problem. We tried to solve this FAQ retrieval problem through three primary approaches. Results from best-performing approaches are taken to form an ensemble method. All three of our approaches are able to convert Latin script in user input query to Devanagari script. We used the indic-trans library for the transliteration (Bhat et al., 2015).

4.1 Dependency Tree Pruning (DTP)

A dependency parse tree was created for the given sentence, and we extracted all the important keywords by pruning the tree using handcrafted rules. Stanza library is used to extract shallow features like Part-Of-Speech (POS) tags and create the dependency tree for Hindi language (Qi et al., 2020). Tree pruning is done in the following three steps:

I. Advice Removal: In the dependency tree, if

[5] easyhindityping.com

[6] We will be releasing a subset of the ASHA-FAQ database to show the working of our chatbot. Full database will be released in future research work.

[7] inshorts.com/hi/read

[8] We also experimented with curated negatives i.e. negative headline = headline (h_j^p) having the highest keyword overlap with h_i where $i \neq j$. It didn't improve the results.

any children of the root node contain words like सलाह (advice) or इलाज (treatment), or if the root node is an inflection of the Hindi word कर (do) and has a child as चाहिए (should) or क्या (what); then the child with the maximum number of descendants is made the new root, and the original root along-with rest of its children are pruned from the tree.

II. Node removal: After a manual analysis of many dependency trees, we inferred that some nodes with specific dependency relations do not contribute to the underlying meaning of the query. The chosen dependency relations were: *dep*, *displocated*, *discourse*, *expl*, *cc*, *case*, *aux*, *aux:pass*, and *mark*. Hence, the nodes connected to the dependency tree with these relations are removed.

III. Compound merging: In the Hindi language, some actions are expressed through a pair of verbs called compound verbs. For eg: रैप करना (wrap doing) here the first verb is the verb stem, and the second verb is a container for inflections like gender, number, and tense. In the compound merging step, all the compound verbs are reduced to their verb stems only. We used the dependency relation called *compound* to identify the compound verbs.

Since the Hindi language generally follows the subject-object-verb paradigm, post-order traversal was used to extract the words from the pruned dependency tree. It is done to make the extracted sentence more readable. Lemmatization is done on the words to remove the inflections during the traversal.

Using the DTP method, we extracted the keywords for every question (Q_i) in the ASHA-FAQ database. Precision and recall between the user query (q) and Q_i is calculated by comparing the overlap between their keywords. We use $F-measure(q, Q_i)$ as the comparison metric, representing the sentence similarity score between q and the i^{th} question in the database (Q_i).

4.2 Sentence-pair Paraphrasing Classifier (SPC)

The notion is to train a deep learning model to predict a score representing the extent to which the given sentence-pair conveys the same information. The predicted score from the classifier is taken as the sentence similarity score for a given sentence-pair. If two sentences in a given sentence-pair convey identical information, then the trained model is supposed to predict a value closer to one. We fine-tune a pretrained multilingual-transformer-encoder (or simply *encoder* henceforth) responsible for generating d-dimensional embeddings for the given sentence-pair. The embeddings are fed to a Feed-Forward Neural Network (FFNN) with a single output node to predict the sentence-similarity score. Earlier works have shown the superiority of fine-tuned *encoders* for paraphrase detection tasks in Hindi sentences under the IndicGLUE benchmark (Kakwani et al., 2020; Venkatesh et al., 2022). We fine-tuned our SPC on the Inshorts dataset and *AshaQs* dataset using the Huggingface library (Wolf et al., 2020).

4.3 Cosine Similarity (COS)

We used different *encoder*s to obtain a d-dimensional vector representation of q and Q_i, as $E(q)$ and $E(Q_i)$, respectively. We used a pre-trained *encoder* from the SentenceTransformer library (Reimers and Gurevych, 2020) to obtain the vector representation of sentences. The traditional cosine similarity between $E(q)$ and $E(Q_i)$ represents the sentence similarity score between q and Q_i.

Ensemble method ($\mathcal{E}$)

The DTP methodology was selected due to its interpretability, in contrast to the SPC and COS methodologies, which have demonstrated remarkable results in sentence similarity tasks. Additionally, we present an ensemble technique that generates sentence similarity scores by leveraging the outputs of the aforementioned three primary methodologies.

For every input query, each approach above produced a list of the most similar QnA pair from the ASHA-FAQ database, along with their respective sentence similarity score. Top-k QnA pairs with the highest scores are chosen as the final suggestions for each input query. It was observed that for some input queries, one approach performed better than the rest, whereas it performed worse for some. Hence, an ensemble method is developed to construct another top-k suggestion from the final suggestions of different approaches. The ensemble method adds the scores of repeated suggestions, and top-k suggestions having the highest

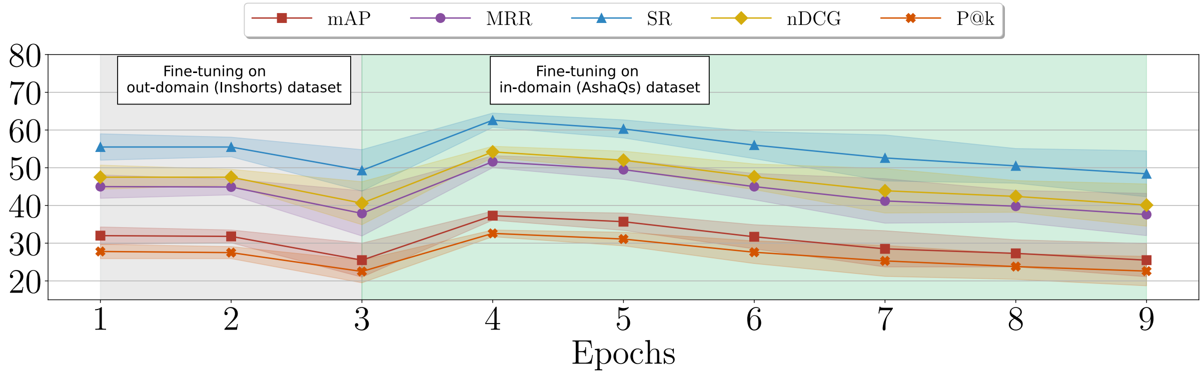

Figure 2: Performance of fine-tuned SPC on the *hold-out test-set* with ten different random seeds. A random seed is responsible for weight initialization in linear layers and the data shuffling between training-testing sets before fine-tuning. The fine-tuned SPC produces top-k QnA suggestions for a given user query (q) where k equals 3. The solid line and the shaded region represent the mean and standard deviation, respectively. The figure depicts the rise in performance of SPC approach when it is fine-tuned on an in-domain data for a single epoch.

	DTP	DTP_{q-e}	SPC	SPC_{+A}	SPC_{q-e}	COS	COS_{q-e}	$\mathcal{E}$
mAP	30.5	35.1	39.4	31.1	39.1	26.5	27.9	**45.3**
MRR	42.6	48.5	54.6	42.2	54.2	38.7	41.0	**61.6**
SR	27.1	59.6	66.2	49.6	64.4	47.7	51.1	**70.3**
nDCG	45.5	51.2	57.1	43.9	56.5	40.8	43.3	**62.5**
P@3	27.1	30.0	34.6	34.6	34.6	22.7	23.9	**34.6**

Table 1: Comparison of all three primary approaches on *hold-out test set* for top-3 suggestions extracted by our chatbot. The ensemble ($\mathcal{E}$) is obtained by taking the best-performing models, highlighted with yellow, from each primary approach. Evidently, the ensemble approach outperforms all the other approaches.

scores are extracted as final suggestions of the ensemble method.

5 Results

It has been observed that, among the top-3 suggestions, DTP gave at least one relevant suggestion only in 27.1% of user queries in the *hold-out test set*. We explored the possible reasons for its failures and found out that the method could not handle the polysemous nature of words. For example, DTP considers शुगर (sugar) and डायबिटीज (diabetes) as entirely different words. However, the two words are interchangeably used in the Indian subcontinent to describe a prevalent disease called *Diabetes mellitus*.

We tried to solve the polysemous word problem by maintaining buckets of such words. Whenever a single word from a bucket is encountered in either q or Q_i, the rest of the words from the bucket are added to the sentence. Expanding the query in such a manner is called *query-expansion (q-e)* in automatic question answering (Ray et al., 2018). It is shown to improve the DTP method by giving relevant suggestions in 59% of the user queries. Table 1 shows the performance boost in DTP due to *q-e* variation. Ablation study highlighting the importance of different pruning strategies in DTP is illustrated in Table 3 of Appendix A.

Multiple *encoders* were used to build the SPC model. It was observed that the *bert-base-multilingual-cased* (*mbert*) *encoder* by Devlin et al. (2018), gave better results than other pre-trained multilingual *encoders*. Moreover, fine-tuning SPC model with three linear layers on top of the *encoder* resulted in the best performance. Since Rogers et al. (Rogers et al., 2020) suggested that early layers of *encoders* contain more syntactic information, we froze the early layers of the *encoder*. We observed more stable results across different random seeds. We first fine-tune the resulting model on the open domain Inshorts dataset and then fine-tune it further on the *AshaQs* dataset in the healthcare domain. We observed that it boosted the performance of SPC on the *hold-out test-set* in the fourth epoch, as shown in Figure 2. Table 1 shows that $q - Q_i$ sentence similarity works better than the $q - Q_iA_i$ similarity, which is aligned with

	$\mathcal{E}_{-COS}$	$\mathcal{E}_{-DTP}$	$\mathcal{E}_{-SPC}$	$\mathcal{E}$
mAP	40.9	40.8	30.3	**45.3**
MRR	56.2	56.5	43.8	**61.6**
SR	66.2	66.2	51.1	**70.3**
nDCG	58.4	58.2	45.5	**62.5**
P@3	34.6	34.6	23.9	**34.6**

Table 2: Results of ablation study on the Ensemble method ($\mathcal{E}$). Th table illustrates that removing any approach ($COS/DTP/SPC$) from the ensemble method results in lower performance.

earlier works (Bhagat et al., 2020; Sakata et al., 2019). Sensitivity of the SPC model with respect to other architectural choices is given in Table 4 of Appendix A.

Calculating sentence similarity score as the cosine distance between the vector representations of two sentences is also effective. We observed that using *paraphrase-multilingual-mpnet-base-v2* as the pretrained *encoder* gave better results than other *encoders* from the SentenceTransformer library. Table 1 shows that using the *q-e* variations on q and Q improved the COS results.

Table 1 shows that the ensemble method $\mathcal{E}$ outperformed all three approaches on the *hold-out test set*. We performed an ablation study to assess the importance of each component of $\mathcal{E}$. The minus sign in the subscript represents the absence of that particular component. For example, if SPC is absent, then it is represented by $\mathcal{E}_{-SPC}$. Table 2 shows that removing any component decreases the performance of $\mathcal{E}$. It was also observed that when the three approaches produced top-5 suggestions, the resulting ensemble method achieved a Success Rate of 73%. Moreover, the chatbot gives a better SR value for user queries with many relevant questions in the ASHA-FAQ database.

The SPC approach majorly dominates the inference time of the ensemble method. It was observed that, with a GPU-enabled server, the ensemble chatbot gives real-time suggestions in 4 seconds and consumes a memory of 2.3 GB on the GPU. However, the chatbot takes a few minutes to generate top-k suggestions without a GPU and consumes a memory of 6.0 GB of RAM.

6 Limitations

In our study, we tested the chatbot on the Hindi database, which humans heavily annotated. Thus, when the database size is enormous, the scalability of the annotation approach is a critical question. Since the questions and answers could be possible in different languages, it will require considerable effort to translate them and, at the same time, preserve their context. In our study, we observed the success ratio of the developed chatbot to be 70% for Hindi queries. However, it is not indicative of its performance in different natural languages.

For a given user query (q), the performance of our best approach for the FAQ-retrieval system is highly dependent on the number of different relevant questions (Q) existing in our ASHA-FAQ database for the given q. Considering the large number of user queries that can be asked in the healthcare field, the small size of our ASHA-FAQ database is a significant reason behind the instances where our method fails to suggest relevant questions (Q) to the user. Moreover, our work does not analyze the quality of answers present in the ASHA-FAQ database. Hence, a user study would be required to analyze the questions' diversity and the answers' quality in our ASHA-FAQ database.

7 Conclusion and Future Work

In this paper, we presented the development of a chatbot to reduce the workload of healthcare professionals for extending informational support regarding maternal and child healthcare concerns in a resource-constrained environment. We followed a FAQ-based model to develop our chatbot using a healthcare database curated in Hindi. Our developed FAQ chatbot can process Hindi user queries written in either the native script of Hindi (Devanagari) or in the native script of English (Latin). We experimented with different FAQ-retrieval methods to extract the most relevant QnA pairs from a FAQ database. We found that the chatbot has the potential to provide relevant QnA pairs for up to 70% queries that our FAQ database can answer. In the future, we plan to evaluate the bot in the wild with healthcare professionals involved.

We plan to evaluate our chatbot with pregnant and postpartum women in a resource-constrained environment to understand the performance of the chatbot in the wild. We also plan to incorporate a healthcare professional to answer questions beyond the chatbot's capacity. The answer obtained from the professional will be further added to the existing QnA database for handling future queries, which would improve the chatbot's success rate over time.

Acknowledgements

Ritwik Mishra would like to express his gratitude towards University Grant Commission (UGC), India, as he is partly supported by the UGC Junior Research Fellowship (JRF) program. Rajiv Ratn Shah acknowledges the partial support provided by the Infosys Center for AI (CAI) and the Center of Design and New Media (CDNM) at IIIT Delhi. We would like to acknowledge the support of the iHub Anubhuti Foundation for supporting this research. We also acknowledge the support of SWACH foundation.

References

Monica Agrawal, Janette Cheng, and Caelin Tran. 2017. What's up, doc? a medical diagnosis bot. *Spoken Language Processing (CS224S), Spring*.

Pranav Bhagat, Sachin Kumar Prajapati, and Aaditeshwar Seth. 2020. Initial lessons from building an ivr-based automated question-answering system. In *Proceedings of the 2020 International Conference on Information and Communication Technologies and Development*, ICTD2020, New York, NY, USA. Association for Computing Machinery.

Irshad Ahmad Bhat, Vandan Mujadia, Aniruddha Tammewar, Riyaz Ahmad Bhat, and Manish Shrivastava. 2015. Iiit-h system submission for fire2014 shared task on transliterated search. In *Proceedings of the Forum for Information Retrieval Evaluation*, FIRE '14, pages 48–53, New York, NY, USA. ACM.

Eric Brill, Susan Dumais, and Michele Banko. 2002. An analysis of the askmsr question-answering system. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP 2002)*, pages 257–264.

Benilda Eleonor V Comendador, Bien Michael B Francisco, Jefferson S Medenilla, and Sharleen Mae. 2015. Pharmabot: a pediatric generic medicine consultant chatbot. *Journal of Automation and Control Engineering*, 3(2).

Jeanne E Daniel, Willie Brink, Ryan Eloff, and Charles Copley. 2019. Towards automating healthcare question answering in a noisy multilingual low-resource setting. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 948–953.

Ashavaree Das and Madhurima Sarkar. 2014. Pregnancy-related health information-seeking behaviors among rural pregnant women in india: validating the wilson model in the indian context. *The Yale journal of biology and medicine*, 87(3):251.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Santiago González-Carvajal and Eduardo C Garrido-Merchán. 2020. Comparing bert against traditional machine learning text classification. *arXiv preprint arXiv:2005.13012*.

Manraj Singh Grover, Pakhi Bamdev, Yaman Kumar, Mika Hama, and Rajiv Ratn Shah. 2020. audino: A modern annotation tool for audio and speech.

Yaru Hao, Li Dong, Furu Wei, and Ke Xu. 2019. Visualizing and understanding the effectiveness of bert. *arXiv preprint arXiv:1908.05620*.

Divyanshu Kakwani, Anoop Kunchukuttan, Satish Golla, NC Gokul, Avik Bhattacharyya, Mitesh M Khapra, and Pratyush Kumar. 2020. Indicnlpsuite: Monolingual corpora, evaluation benchmarks and pre-trained multilingual language models for indian languages. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4948–4961.

Jasmeet Kaur, Asra Sakeen Wani, and Pushpendra Singh. 2019. Engagement of pregnant women and mothers over whatsapp: Challenges and opportunities involved. In *Conference Companion Publication of the 2019 on Computer Supported Cooperative Work and Social Computing*, CSCW '19, page 236–240, New York, NY, USA. Association for Computing Machinery.

Govind Kothari, Sumit Negi, Tanveer A Faruquie, Venkatesan T Chakaravarthy, and L Venkata Subramaniam. 2009. Sms based interface for faq retrieval. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 852–860.

Mamta Mittal, Gopi Battineni, Dharmendra Singh, Thakursingh Nagarwal, and Prabhakar Yadav. 2021. Web-based chatbot for frequently asked queries (faq) in hospitals. *Journal of Taibah University Medical Sciences*, 16(5):740–746.

Basant Potnuru et al. 2017. Aggregate availability of doctors in india: 2014–2030. *Indian journal of public health*, 61(3):182.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. Stanza: A Python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*.

Khyathi Raghavi, Manoj Chinnakotla, and Manish Shrivastava. 2015. " answer ka type kya he? " learning to classify questions in code-mixed language.

Santosh Kumar Ray, Amir Ahmad, and Khaled Shaalan. 2018. A review of the state of the art in hindi question answering systems. *Intelligent Natural Language Processing: Trends and Applications*, pages 265–292.

Nils Reimers and Iryna Gurevych. 2020. Making monolingual sentence embeddings multilingual using knowledge distillation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2020. A primer in bertology: What we know about how bert works. *Transactions of the Association for Computational Linguistics*, 8:842–866.

Shriya Sahu, Nandkishor Vasnik, and Devshri Roy. 2012. Prashnottar: a hindi question answering system. *International Journal of Computer Science & Information Technology*, 4(2):149.

Wataru Sakata, Tomohide Shibata, Ribeka Tanaka, and Sadao Kurohashi. 2019. Faq retrieval using query-question similarity and bert-based query-answer relevance. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1113–1116.

Satoshi Sekine and Ralph Grishman. 2003. Hindi-english cross-lingual question-answering system. *ACM Transactions on Asian Language Information Processing (TALIP)*, 2(3):181–192.

Gopalakrishnan Venkatesh, Abhik Jana, Steffen Remus, Özge Sevgili, Gopalakrishnan Srinivasaraghavan, and Chris Biemann. 2022. Using distributional thesaurus to enhance transformer-based contextualized representations for low resource languages. In *Proceedings of the 37th ACM/SIGAPP Symposium on Applied Computing*, pages 845–852.

Joseph Weizenbaum. 1966. Eliza—a computer program for the study of natural language communication between man and machine. *Commun. ACM*, 9(1):36–45.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of EMNLP 2020: System Demonstrations*, pages 38–45, Online. ACL.

Deepika Yadav, Anushka Bhandari, and Pushpendra Singh. 2019a. Leap: Scaffolding collaborative learning of community health workers in india. *Proc. ACM Hum.-Comput. Interact.*, 3(CSCW).

Deepika Yadav, Kirti Dabas, Prerna Malik, Anushka Bhandari, and Pushpendra Singh. 2022. “should i visit the clinic”: Analyzing whatsapp-mediated online health support for expectant and new mothers in rural india. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, CHI ’22, New York, NY, USA. Association for Computing Machinery.

Deepika Yadav, Prerna Malik, Kirti Dabas, and Pushpendra Singh. 2019b. Feedpal: Understanding opportunities for chatbots in breastfeeding education of women in india. *Proc. ACM Hum.-Comput. Interact.*, 3(CSCW).

Deepika Yadav, Prerna Malik, Kirti Dabas, and Pushpendra Singh. 2021. Illustrating the gaps and needs in the training support of community health workers in india. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, CHI ’21, New York, NY, USA. Association for Computing Machinery.

Deepika Yadav, Pushpendra Singh, Kyle Montague, Vijay Kumar, Deepak Sood, Madeline Balaam, Drishti Sharma, Mona Duggal, Tom Bartindale, Delvin Varghese, and Patrick Olivier. 2017. <i>sangoshthi</i>: Empowering community health workers through peer learning in rural india. In *Proceedings of the 26th International Conference on World Wide Web*, WWW ’17, page 499–508, Republic and Canton of Geneva, CHE. International World Wide Web Conferences Steering Committee.

A Appendix

	mAP	MRR	SR	nDCG	P@3
DTP_{q-e}	**35.1**	**48.5**	**59.6**	**51.2**	**30.0**
-any pruning	25.5	37.3	45.2	39.1	21.3
-advice removal	30.3	43.1	54.4	46.0	27.0
-node removal	28.4	40.9	53.7	44.2	27.0
-compound merging	31.1	44.4	55.1	47.0	27.0

Table 3: An ablation of different pruning strategies in the DTP method. In absence of any pruning strategy, simple lemmatization, stop-word removal, and token matching is performed.

Fine-tuning data	**Pretrained Encoder**	**Linear Layers**	**Frozen Layers**	**Best Epoch**	**SR spread ($\mu \pm \sigma$)**	**Best SR**
Inshorts 3 epoch AshaQs 6 epoch	mbert-cased	3	embedding, layer 0	4	62.6 ± 1.9	66.2
AshaQs 4 epoch	mbert-cased	3	embedding, layer 0	1	▼ 62.4 ± 3.7 ▲	67.0 ▲
Inshorts 4 epoch	mbert-cased	3	embedding, layer 0	1	▼ 55.5 ± 3.5 ▲	60.0 ▼
Inshorts 1 epoch **AshaQs 1 epoch**	mbert-cased	3	embedding, layer 0	2	▲ 64.3 ± 2.4 ▲	67.8 ▲
Inshorts 2 epoch **AshaQs 1 epoch**	mbert-cased	3	embedding, layer 0	3	▲ 62.9 ± 1.5 ▼	65.9 ▼
Inshorts 4 epoch **AshaQs 1 epoch**	mbert-cased	3	embedding, layer 0	5	▼ 61.0 ± 2.9 ▲	64.1 ▼
Inshorts 3 epoch AshaQs 6 epoch	**xlm-roberta**	3	embedding, layer 0	4	▼ 61.3 ± 2.5 ▲	65.2 ▼
Inshorts 3 epoch AshaQs 1 epoch	**indic-bert**	3	embedding, layer 0	4	▼ 5.9 ± 0.8 ▼	7.0 ▼
Inshorts 3 epoch AshaQs 1 epoch	**mbert -uncased**	3	embedding, layer 0	4	▼ 60.0 ± 4.3 ▲	65.9 ▼
Inshorts 3 epoch AshaQs 1 epoch	mbert-cased	**1**	embedding, layer 0	4	▼ 57.8 ± 2.5 ▲	61.9 ▼
Inshorts 3 epoch AshaQs 1 epoch	mbert-cased	**2**	embedding, layer 0	4	▼ 60.6 ± 1.7 ▼	63.0 ▼
Inshorts 3 epoch AshaQs 1 epoch	mbert-cased	**4**	embedding, layer 0	4	▼ 61.3 ± 2.5 ▲	64.8 ▼
Inshorts 3 epoch AshaQs 1 epoch	mbert-cased	3	**embedding**	4	▼ 61.3 ± 3.4 ▲	66.7 ▲
Inshorts 3 epoch AshaQs 1 epoch	mbert-cased	3	**embedding, layer 0, 1**	4	▼ 61.3 ± 2.4 ▲	63.0 ▼
Inshorts 3 epoch AshaQs 1 epoch	mbert-cased	3	**embedding, layer 0, 1, 2**	4	▼ 61.1 ± 2.2 ▲	63.7 ▼
Inshorts 3 epoch AshaQs 1 epoch	mbert-cased	3	**half bert**	4	▼ 61.9 ± 2.1 ▲	64.4 ▼
Inshorts 3 epoch AshaQs 1 epoch	mbert-cased	3	**nothing**	4	▼ 61.5 ± 2.1 ▲	65.6 ▼

Table 4: Sensitivity of SPC approach due to different architectural choices. Each experiment is run with ten random seeds. For the sake of brevity, we have chosen the Success Ratio (SR) to represent the overall performance since, in our experiments, it acts as an upper bound of all the evaluation metrics. The first row of the table contains the architectural choices of the best SPC approach. Red-colored triangles (▲/▼) represent a drop in performance as compared to the best model. Note: increased standard deviation (σ) indicates more numerical instability, hence worse performance. Since no row contains all green colored triangles, it shows that the configuration of first row is the best configuration.