A Framework For Automatic Question Answering in Indian Languages

By

Ritwik Mishra

With the supervision of

Dr. Rajiv Ratn Shah, IIIT Delhi
Prof. Ponnurangam Kumaraguru, IIIT Hyderabad

Indraprastha Institute of Information Technology Delhi

July, 2022

Abstract

The distribution of research efforts done in the field of Natural Language Processing (NLP) has not been uniform across all natural languages. It has been observed that there is a significant gap between the development of NLP tools in Indic languages (indic-NLP), and in European languages. We aim to explore different directions to develop an automatic question answering system for Indic languages. We built a FAQ-retrieval based chatbot for healthcare workers and young mothers of India. It supported Hindi language in either Devanagri script or Roman script. We observed that, in our FAQ database, if there exists a question similar to the query asked by the user, then the developed chatbot is able to find a relevant Question-Answer pair (QnA) among its top-3 suggestions 70% of the time. We also observed that performance of our chatbot is dependent on the diversity in the FAQ database. Since database creation requires substantial manual efforts, we decided to explore other ways to curate knowledge from raw text irrespective of domain.

We developed an Open Information Extraction (OIE) tool for Indic languages. During the preprocessing, chunking of text is performed with our fine-tuned chunker, and the phrase-level dependency tree was constructed using the predicted chunks. In order to generate triples, various rules were handcrafted using the dependency relations in Indic languages. Our method performed better than other multilingual OIE tools on manual and automatic evaluations. The contextual embeddings used in this work does not take syntactic structure of sentence into consideration. Hence, we devised an architecture that takes the dependency tree of the sentence into consideration to calculate Dependency-aware Transformer (DaT) embeddings.

Since the dependency tree is also a graph, we used Graph Convolution Network (GCN) to incorporate the dependency information into the contextual embeddings, thus producing DaT embeddings. We used a hate-speech detection task to evaluate the effectiveness of DaT embeddings. Our future plan is to evaluate the applicability of DaT embeddings for the task of chunking. Moreover, the broader aim for the future is to develop an end-to-end pronoun resolution model to improve the quality of triples and DaT embeddings. We also aim to explore the applicability of all our works to solve the problem of long-context question answering.

Contents

Chapter 1

Introduction

The field of Artificial Intelligence (AI) has progressed to a great extent in the last decade. We have witnessed the growing applicability of AI models ranging from movie scene generation starring deceased actors [1] to booking appointments over a telephonic conversation while pretending to be a human [2]. The later application is an example of a Natural Language Process (NLP) tool; it is a branch of AI that deals with computational processing of human languages in text or speech modality. It is due to extensive research in the field of NLP that we are able to witness remarkable achievements such as Machine Translation, Speech Recognition, Fake News Detection, Automatic Lip Reading and many more. The task of developing NLP tools becomes more challenging due to ambiguity in natural languages. In our work, we have focused on the text modality of natural languages since it has been conjectured that written language has played an important role in cognitive development of humankind [3].

1.1 Why Indian Languages?

Indian subcontinent is a quintessential example of what linguists describes as a *linguistic area* due to existence of many languages from different families in this geographical region [4]. Today many Indian languages are among the top-20 most spoken languages in the world [5, 6]. The number of Indians consuming or producing web content in their native language is increasing with time [7]. Despite being spoken by billions of people, Indic languages are considered low-resource due to the lack of annotated resources and automated systems available for them [8]. As a result, the development of Natural Language Processing (NLP) tools for Indian languages (Indic-NLP) has been much slower as compared to some the high-resource European languages[1] (like *French* and *Italian*) with lesser number of speakers. Hence the primary motivation behind working with low-resource Indian languages is

[1] `https://github.com/sebastianruder/NLP-progress`

to initiate research in the field of Indic-NLP, leading to the development of further resources in this domain.

1.2 Question Answering

Automatic Question Answering (QnA) is a downstream task in NLP, and it started in early 1960s [9]. Figure 1.1 illustrates different types of QnA systems. However, the development of Conversational Agents (CAs) as QnA has recently gained popularity among NLP researchers [10]. Conversational Agents (also known as chatbots[11]) are the systems designed to interact with a human, through text or audio, in order to imitate a natural conversation between two humans through a series of question answering sessions. One of the most common application of CAs is in the domain of healthcare and well-being [12, 13]. Dissemination of information and awareness regarding healthcare becomes of utmost importance in the remote places of India. Due to inequality in healthcare access across urban and rural parts of India, people in the rural areas have limited availability of healthcare professionals (such as doctors, nurses, midwife), and hence suffer from low access to healthcare [14, 15].

In resource-constrained environments, pregnant and postpartum women face challenges in seeking healthcare information due to barriers such as limited time with the healthcare professionals, lack of authentic information on health issues. As a result, they rely on non health professionals such as mothers, mothers-in-law for health information which might lead to health related misconceptions and health issues [14, 16, 17]. This presents an opportunity to explore ways to extend informational support to pregnant and postpar-

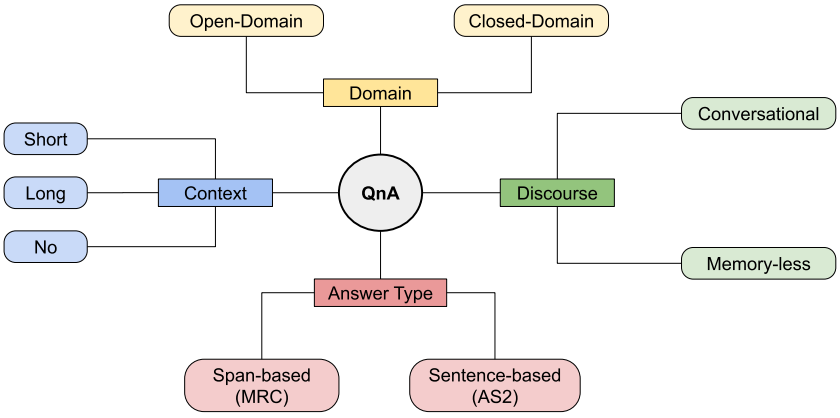


Figure 1.1: Different types of Automatic Question-Answering systems (QnA) depending upon the domain, question-context, answer-type, and discourse in questions.

tum women in their local languages while reducing workload of doctors by means of engaging with emerging technology such as chatbots to support the question-answering. Chatbots have been found to have a potential to act as a first point of contact for women seeking answers for maternal and child healthcare related queries, especially in resource-constrained environments [18]. A chatbot can be designed in the following three ways: (a) Rule-based approach (b) Generative-based approach, and (c) Retrieval-based approach [19]. In this report, we have emphasized on the third approach.

1.2.1 Retrieval-based Chatbot

Given a database of FAQ and their corresponding answers, the process of retrieval deals with extracting k most relevant question-answer pairs (QA) from the database for a real-time user query [20]. In the absence of huge amount of labeled data, maintaining a FAQ database is a popular method to represent domain-specific knowledge [21]. It has been observed that retrieval-based approaches yields more fluent responses as compared to other approaches [22, 23, 24].

The chatbot provides top-k most relevant QnA pairs in response to a healthcare query (q) using our curated database of QnA pairs with answers vetted by healthcare professionals. This approach of extracting relevant and verified QnA pairs prevents the chances of chatbot providing incorrect answers which could result in severe health issues. A sentence similarity score is calculated to determine the extent of relatedness between a user-query (q), and each row in the FAQ database. Figure 1.2 illustrates internal working of a retireval-based chatbot. The output consists of a FAQ database sorted by the sentence similarity score in decreasing order. Top-k entries from the sorted database are extracted and sent to the user. Earlier works have used top-3 and top-5 most relevant QA pairs[25].

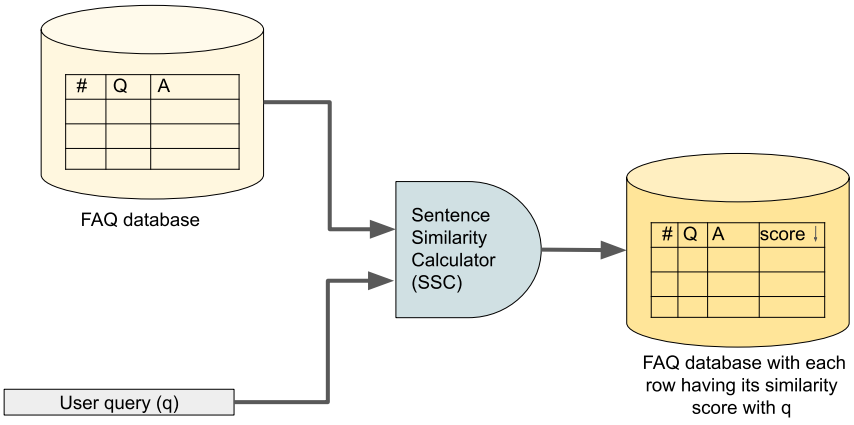


Figure 1.2: Underlying architecture of retrieval-based FAQ chatbot.

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) [20]. 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. However, the retrieval-based approaches are limited to extracting information from an indexed database which has to be curated manually. Such approaches are not able to incorporate the knowledge from long documents containing raw sentences. The transformer-based approaches have a limitation on the size of question-context [26]. Long-context QnA has been previously employed for answering queries from long research articles [27]. Moreover, such systems can be proved useful for developing automated customer support or providing healthcare-related information to the underprivileged in regional languages. In order to retrieve information from a text dump of raw sentences, a knowledge-graph has to constructed from the unstructured text. Open Information Extraction (OIE) tools could be used to extract facts from raw sentences in different domains.

1.3 Open Information Extraction

The concept of Information Extraction (IE) was introduced in the mid-1960s. It deals with extracting structured facts from unstructured text written in a natural language [28]. In IE, most relevant facts are extracted from the text of the documents, whereas in Information Retrieval (IR), most relevant documents are extracted from a document database [29]. Extraction of informative facts irrespective of the text domain is called Open Information Extraction (OIE). A standard convention to represent facts is through triples $<$*head, relation, tail*$>$ where *relation* denotes the link between the two entities, *head* and *tail*. For example, consider the sentence *PM Modi to visit UAE in Jan marking 50 yrs of diplomatic ties*, one of the possible meaningful triple would be $<$*PM Modi*, *to visit*, *UAE*$>$.

The biggest strength of OIE tools is their ability to extract triples from large amounts of texts in an unsupervised manner [30]. OIE also serves as an initial step in building or augmenting a knowledge graph out of an unstructured text [31, 32]. While Relation Extraction has been found to be the most common application of OIE, other fields that benefit from OIE advancements are Ontology Construction, and Fact-Checking [33]. Moreover, many works have used OIE tools to solve downstream applications like Question-Answering [34], Text Summarization [35], and Entity Linking [36].

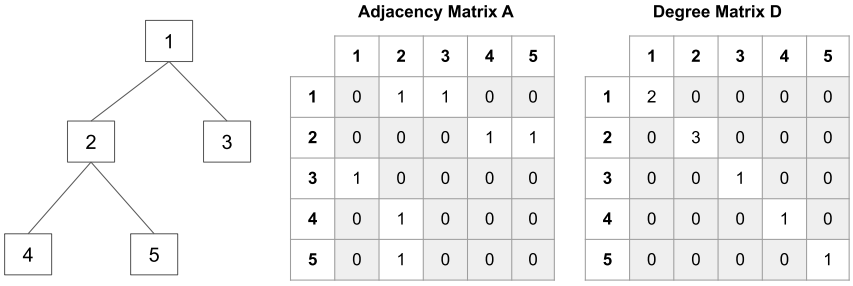
A triple can be extracted in many different ways depending on the word-order constraints in the given natural language and the expected level of details in the triples. Consider the sentence, *John sliced an apple with a*

knife. Two possible ways to extract facts from this sentence are: (i) <*John, sliced, an apple with a knife*> (ii) <*John, sliced, an apple*>, and <*John, sliced, with a knife*>. Both ways represent the same fact but with different levels of detail. In the case of languages with free word order, like Hindi [37], one fact can be represented by many permutations of the elements of a triple. For example, both the following triples <राम ने, खाया, एक सेब> [<*rAm ne*, *khAya*, *ek seb*>][2] and <एक सेब, खाया, राम ने> [<*ek seb*, *khAya*, *rAm ne*>] represents the same information as the following English triple: <*Ram*, *ate*, *an apple*>. However, since the Hindi language uses postpositions instead of prepositions [38], some word permutations in a triple are prohibited in Hindi, whereas they are permissible in English. For example, consider the following Hindi sentence, वह भारत के राष्ट्रपति थे [*veh bhArat ke rAshtrpati the*] (*He was the president of India*). The following two English triples are permitted since they are equally meaningful (i) <*He*, *was the president*, *of India*>, and (ii) <*He*, *was the president of*, *India*>. Whereas the former triple is permitted in Hindi, but the latter triple is prohibited because the Hindi phrase भारत के [*bhArat ke*] represents the English phrase '*of India*' as long as the given two Hindi words appear together and in that order. Hence, separating the postpositional words (such as के [*ke*] (*of*), मे [*me*] (*in*), etc.) from their preceding noun is prohibited in Hindi.

The quality of generated triples is generally evaluated by getting them annotated by native speakers of that language. However, a significant limitation of evaluation methods based on manual annotations is their lack of reproducibility and the time-intensive procedure to perform the annotations. The problem is exacerbated for Indic languages due to the lack of annotators in this field. There are automatic evaluation techniques for triples generated in English [39, 40]. Therefore, methods for automatic evaluation of generated triples in Indic languages also needs to explored.

Extracting triples using dependency parsing is common a method in the prior works [41, 42, 30, 43]. However, dependency parsing does not perform well in capturing Multi-word Entities (MWE) from a given sentence [30]. Grammatical structures such as Complex Predicates (CPs) in Indic languages [44, 45] makes the task of dependency parsing more challenging. Shallow parsing can be used to identify MWEs, and pretrained transformer-based models have shown their superior ability to perform well on shallow parsing tasks [46, 47, 48]. But such contextual embeddings are generated by pretraining on tasks, such as Masked-Language Modelling (MLM) and Next Sentence Prediction (NSP), which do not take the syntactical structure (or word order) of the sentence directly into consideration [49, 50]. As a result, it has been observed that syntactic information is either absent in

[2]Italicized text written in square brackets represents the ITRANS transliteration, whereas the italicized text in round brackets represents the English translation of the preceding Hindi phrase/sentence



	1	2	3	4	5
1	0	1	1	0	0
2	0	0	0	1	1
3	1	0	0	0	0
4	0	1	0	0	0
5	0	1	0	0	0

	1	2	3	4	5
1	2	0	0	0	0
2	0	3	0	0	0
3	0	0	1	0	0
4	0	0	0	1	0
5	0	0	0	0	1

Figure 1.3: An example of an undirected graph with $N = 5$ nodes.

the transformer embeddings or it is not utilized while making the predictions [51]. Therefore, there is a need to explore the possibility of incorporating syntactic (dependency) information in transformer embeddings.

1.4 Dependency-aware Transformer Embedding

Although architectures like Bidirectional Encoder Representations from Transformers (BERT) [49] have shown to generate embeddings containing syntactic information [52, 53], it was observed that BERT representations shows a clear insensitivity to word-order and negations [54]. Earlier works have shown that incorporating dependency information with word embeddings can improve the performance of the resulting model on various downstream tasks such as Named Entity Recognition (NER) [55], Semantic Role Labelling (SRL) [56], and Relation Extraction (RE) [57].

In order to incorporate dependency information with word embeddings, Graph Convolution Networks (GCNs) have been a popular choice in prior works [56, 57, 58]. Kipf and Welling [59] introduced the GCNs that employed different weight matrix (W) for different layer (l). Equation 1.1 illustrates the propagation rule for each layer for graph with N nodes.

$$H^{l+1} = \sigma(\hat{A}H^lW^l);\ where\ \hat{A} = D^{-0.5}\tilde{A}D^{-0.5}\ and\ \tilde{A} = A + I_N \qquad (1.1)$$

The number of layers (L) in a GCN block is an integer hyperparameter where $l \in [0, L-1] \wedge l \in \mathbb{Z}^+$. Figure 1.3 illustrates an example of a simple graph with its corresponding adjacency matrix (A) and degree matrix (D). The transformer embeddings in layer l are represented by H^l. By adding the adjacency matrix (A) and an identity matrix (I_N), a new adjacency matrix $\tilde{A}$ is formed which contains self-connections to each node of the graph. A normalizing matrix ($\hat{A}$) is computed by sandwiching $\tilde{A}$ between $D^{-0.5}$ from both sides. It is done to ensure that embedding of higher-degree nodes does not dominate the graph. Another way to formulate the normalizing matrix

is $\hat{A}_{ij} = \tilde{A}_{ij}/\sqrt{d_i d_j}$ where d_i represents the degree of node i. Notice that $\hat{A}$ is independent of l since the structure of (dependency) graph remains the same throughout.

For batch processing, the input text sequence is padded to N length. If the number of words in the original text was N_w, then the number of pad tokens will be $N_p = N - N_w$. Dependency graph for the padded sequence will be containing a dependency tree with N_w nodes, and N_p isolated nodes. Similarly, the padded tokens will be having only the self-connections in the adjacency matrix ($\tilde{A}$).

The flow of node embeddings between the neighbours in a graph is illustrated in figure 1.4. At each subsequent layer, the embedding of a node (n_i) is calculated by taking a weighted sum of its own embedding and embedding of all its neighboring nodes. Values from the $\hat{A}$ matrix are used as weights for the weighted sum. We can observe from figure 1.4 that embedding of n_3 in layer 1 is calculated using its own embedding and embedding of n_1. It is to be noted that, in figure 1.4, the colored partitions inside a multicolored node are not representative of their contribution while calculating the node embedding. For example, it is true that embedding of n_2 in layer 1 will be calculated using the embedding of n_4 (green), n_5 (blue), n_1 (red), and its own embedding (orange) from layer 0; but their contribution will not be as per their partitions shown in the figure. Values in $\hat{A}$ matrix will be responsible for their contribution. The initial embeddings from a pretrained transformer encoder (H^0) are fed to a GCN block, and after going through L layers we get Dependency-aware Transformer (DaT) embeddings.

Each node in a dependency graph represents a word, and the edge between two nodes represents the dependency relation between the correspond-

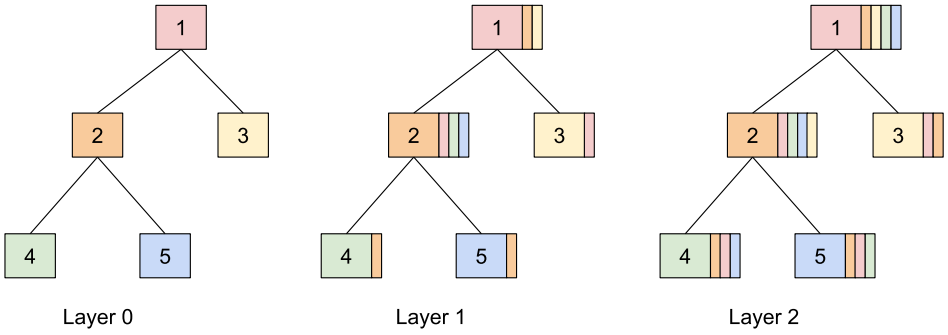


Figure 1.4: Propagation of node embeddings through different layers. Initial embedding (H^0) of each node is highlighted with a different color in Layer 0. In the subsequent layers, any node with multiple colors indicates that its embedding is calculated using node embedding of the corresponding colors in Layer 0. Note: tile area is not representative of the contribution.

ing two words. However, pretrained transformer encoders (or simply *encoders* henceforth) generates contextual embeddings at subword token level. Hence subword level contextual embeddings are to be transformed into word level contextual embeddings. For Named Entity Recognition (NER) task, Devlin et al. [60] solved this problem by taking the embedding of first subword token as a representative of the entire word. There are other methods that take last subword token embedding instead and show better results[3].

1.4.1 Hate-speech detection

In order to evaluate the effectiveness of DaT embeddings, a downstream task is chosen to train models with and without DaT embeddings. We have selected HateSpeech detection as the downstream task due to abundance of annotated data available in Indic languages for the task [61, 62, 63, 64]. However, we believe that judging the effectiveness of DaT embeddings through evaluation metric (accuracy/F1-score) of the downstream task is not enough. Hence, we decided to use different methods that can explain the predictions of a fine-tuned model. The following three methods were used to explain the predictions:

1. Layer-wise Relevance Propagation (LRP) [65]: In LRP, the relevance values are backpropagated from output nodes to input nodes. The prediction values at the output nodes are treated as their respective relevance values. Figure 1.5 illustrates the flow of relevance values in the Feed-Forward Neural Network (FFNN) responsible for classification.

2. Local Interpretable Model-Agnostic Explanations (LIME) [66]: The values of input features are slightly modified to observe its impact on the predictions. If tweaking a feature changes the prediction drastically then the feature is considered as important. In case of *encoders*, the input features to be modified are the `token_ids` of each subword token from the text. The explanations of LIME are only locally faithful, not globally.

3. Shapley Additive Explanations (SHAP) [67]: It considers each word as an active contributor the final predicted value of the text. Shapely value is used to determine the contribution, and hence importance, of each word in calculating the predicted value. It involves masking different word-spans (using a special mask token) in the text and observing the change in the predicted value.

By explainable predictions, the effectiveness of DaT embeddings is to be evaluated by manually annotating a small set of text sequences with

[3] `https://github.com/soutsios/pos-tagger-bert/`

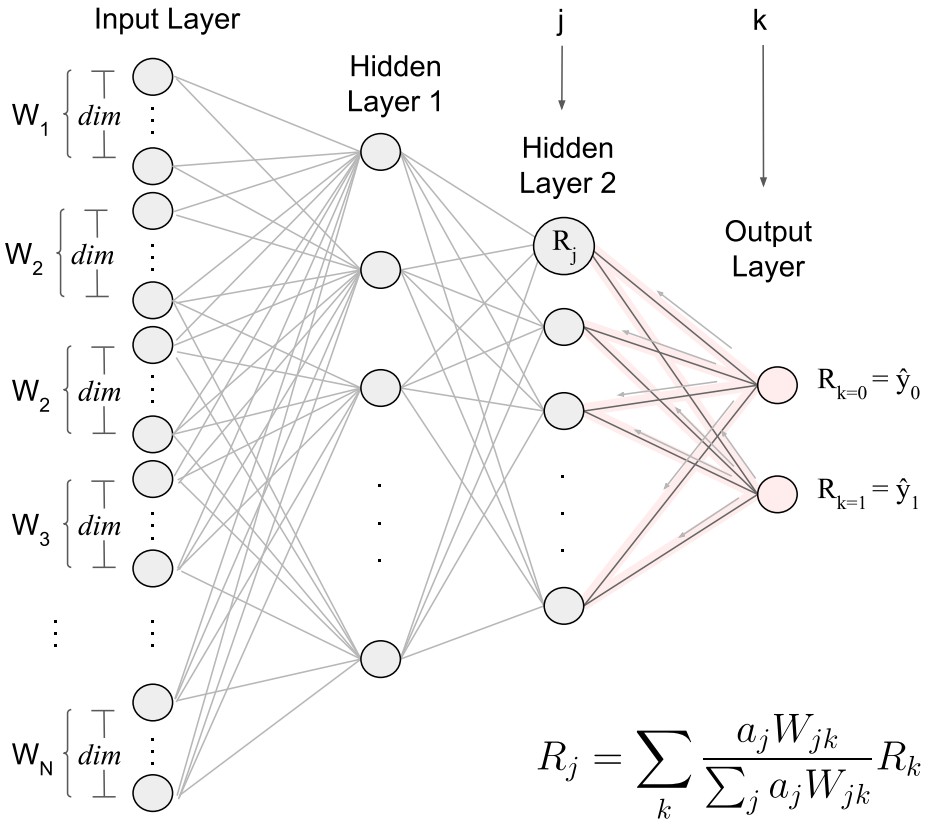


Figure 1.5: Relevance values are backpropagated from the output layer to the input layer in the Feed-Forward Neural Network (FFNN) responsible for classification. Contextual embedding of all the words ($\mathbb{R}^{|N \times dim|}$) is flattened and fed to the classification network. Relevance for a word is calculated by adding the relevance of its corresponding nodes in the input layer

their corresponding prediction and prediction explanations. We hypothesize that, while predicting a label, our HateSpeech detection model with DaT embeddings will be focusing on actual hateful words/phrases better than the HateSpeech detection model without DaT embeddings.

In a multi-sentence text, a single dependency graph will be constructed containing dependency tree of all the sentences. Figure 1.6 shows an example of a dependency graph and its corresponding $\tilde{A}$ $(= A + I_N)$ matrix for a multi-sentence text in Hindi. In figure 1.6, it is observed that the dependency trees of the two sentences remain disconnected to each other in the $\tilde{A}$ matrix despite the same discourse entity being present in the two sentences. A discourse entity can be thought of like a '*topic of discussion*'. Therefore, a coreference resolution mechanism is required to link mentions from different sentences talking about the same discourse entity. Prior works have shown that coreference resolution mechanism will also be helpful in generat-

	राम	अयोध्या	लौटे	।	वह	राजा	बने	।	[PAD]	[PAD]
राम	1		1							
अयोध्या		1	1							
लौटे	1	1	1	1						
।			1	1						
वह					1		1			
राजा						1	1			
बने					1	1	1	1		
।							1	1		
[PAD]									1	
[PAD]										1

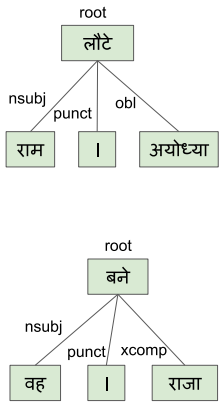


Figure 1.6: Dependency graph (right) and $\tilde{A}$ $(= A + I_N)$ matrix (left) for a multi-sentence Hindi text, राम अयोध्या लौटे। वह राजा बने। which translates to *Ram returned (to) Ayodhya. He became (a) king.* The text is padded to a length of $N = 10$. Empty grey cells in $\tilde{A}$ represents zero values.

ing meaningful triples for an OIE task [68, 69, 70, 71].

1.5 Coreference Resolution

Any natural language sentence is composed of two different mentions i.e. entities and events. An event represents the underlying action of a sentence whereas an entity represents the attributes of action such as participants in the action, location of the action, nature of the action, etc. Copular verbs are a special category of events where the action is to establish an equivalence. For example, consider the following sentence *Shah Rukh Khan is an Indian actor*, here the event (*is*) represents the action of establishing the equivalence between the two entities *Shah Rukh Khan* and *an Indian actor*. In any natural language, two expressions are said to corefer if a consumer of the information (listener or reader or viewer) believes that the two expressions refer to the same real-world[4] concept. The process of resolving coreferences and forming a unique coreference chain for each real-world concept is called coreference resolution.

Event (coreference) resolution has been more challenging as compared to entity resolution even in high-resource languages [72, 73]. Therefore, we have focused on entity resolution in this report. The real-world concept to which a mention refers to is called its referent. If an entity mention refers to an entity that has been introduced earlier in the discourse then it is called anaphora or anaphoric mention whereas the prior entity mention to which

[4]For simplicity, we make an assumption that all mentions (be it fictional/virtual/mythical/imaginary etc) are real-world first

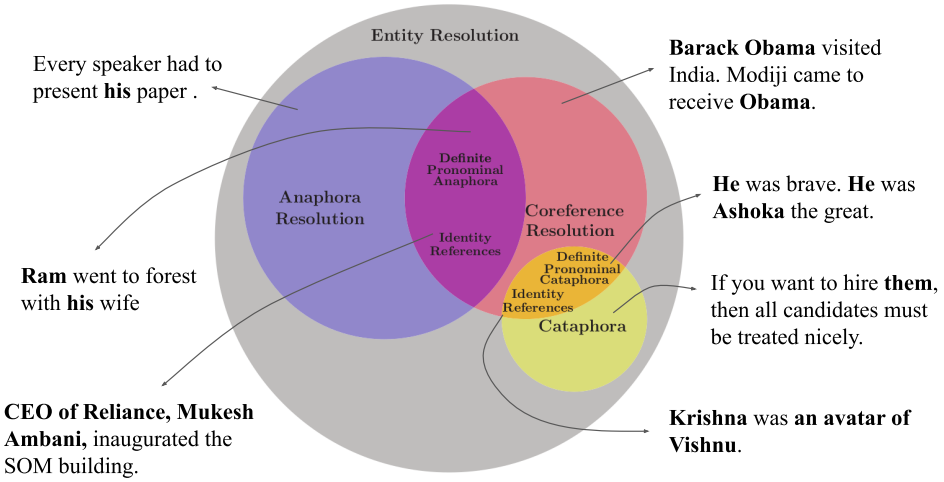


Figure 1.7: Categorization of the entity resolution process [74]. Entity mentions are highlighted in **bold** text.

the anaphoric mention corefers is called its antecedent. For example, consider these two sentences *'Mahatma Gandhi started the Dandi March in 1930. He was arrested later.'*, here the anaphoric mention is *He* and its antecedent is *Mahatma Gandhi*. The referent to both the mentions (*He* and *Mahatma Gandhi*) will be a real-world concept who was a person named *Mohandas Karamchand Gandhi*. It is not necessary for an anaphoric expression to refer to a previously mentioned entity. An anaphoric expression may or may not be a pronoun. When an entity mention refers to an entity that occurs later in the discourse then it is called cataphora. Figure 1.7 illustrates the different ways by which two entity mentions can corefer.

The first step in coreference resolution is to detect mentions. After the mentions are identified, one of the following three architectural choices are used to build the resolution model [75]: (a) Mention-pair architecture: a pair of identified mentions are fed to binary classification model which is responsible for predicting whether the two mentions corefer or not, (b) Mention-rank architecture: calculates the probability of two mentions coreferring with respect to other mentions in the text, and (c) Entity-based models: instead of working with pair of mentions, it calculates the probability of the given mention being a member of a previously identified coreference chain (mention clusters). Rahman and Ng [76] has pointed out that mention-rank architecture can be transformed into an entity-based model. Mention-pair architectures are known to suffer from imbalanced classes in the training data [75]. We have focused on mention-rank architectures in this report since they are more popular than entity-based models [72], and they have shown superior performance as compared to mention-pair models [77].

There are some specific types of coreference which are difficult to resolve. For example, in discourse deixis [78] the antecedent to an anaphora is a discourse segment i.e. an entire clause. Some languages even have a concept of *zero anaphora* [79] where anaphoric expression is not written but implied. We intend to focus on pronomial anaphora resolution in Indian languages since pronomial mentions are the most common anaphoric expressions in a natural language [74].

1.6 Thesis Statement

A broad thesis statement for our work is to explore the possibility to develop a framework for Automated Question-Answering (QnA) in Indian languages with the help of multiple supporting tasks like Open Information Extraction (OIE) and Pronoun Resolution. Moreover, we intend to improve the contextual embeddings of transformer-based architectures and investigate their applicability in the QnA framework.

Chapter 2

Literature Review

2.1 Conversational Agents

Research studies in Human Computer Interaction (HCI) have been conducted to understand the feasibility and acceptability of Automatic Question Answering Systems such as Conversational Agents (or chatbots) for different purposes such as facilitating group discussion, encouraging participation from inactive members in a group, fostering self-disclosure of people on sensitive topics, counseling teenage victims of online sexual exploitation, and facilitating information sharing and fostering social connection within a co-living space [80, 81, 82, 83, 84]. Interactive voice response (IVR) based interventions have also been explored to provide information to pregnant women [85, 86].

Earlier works on developing chatbots in healthcare using AI started with user query reformulation and using knowledge from search engines [87]. They were made for English language, and due to scarcity of resources, the same techniques could not be used for Hindi speakers. Kothari et al. (2009) aimed to develop a FAQ retrieval system for unstructured English language written as a shorthand for Short Messaging Service (SMS) by Indian population [88]. It relied on character level features to calculate the sentence similarity scores. Initial works on building a QA system for Hindi language were restricted to exploiting information from shallow speech features like POS tags [89]. In constructing QA system for English-Hindi code-switched languages, word-level translation of code-switched queries to English queries was a common practice due to lack of resources in Hindi language [90, 91, 92]. Such approaches fails to generalize because Hindi-to-English word-level translations are highly dependent on the position of the Hindi word in the sentence [93].

Previously cross-lingual word embeddings has been used to solve a healthcare QA system in low-resource African languages[94]. Empirically, it has been proven that machine learning models using BERT *embeddings* outperform many other traditional AI models on various tasks[95, 96]. Earlier

works have shown the efficiency of BERT-based models in measuring sentence similarity for FAQ retrieval tasks[25, 20].

2.2 Triple Generation

Earlier works have used a combination of shallow parsing and hand-crafted rules to extract meaningful entities from English language text [97]. However, the field of OIE in English was started by Banko et al. [98] by introducing the TextRunner architecture. It was a single-pass triple extractor that needed no hand-labeled training data. Basic shallow parsing, such as Part of Speech (POS) tagging and chunking, was used to capture the entities in the sentence. Later, researchers took the work forward by using Semantic Role Labelling (SRL) instead of shallow parsing [99]. Their analysis suggested that even though SRL gave more informative entities (or arguments), shallow parsing-based methods were more time-efficient. The problem of incoherent relation extraction was tackled by Fader et al. [100] by using simple lexical constraints for relations.

Mausam et al. [101] explored a different direction for capturing relations mediated by non-verbal phrases like *'is the president of'*. They used hand-crafted rules and dependency parsing to develop OLLIE, which extracted triples from English sentences. OLLIE was found to be performing at par with an SRL-based triple extractor. While most of the works dealt with extracting facts in the form of triples, KrakeN was developed to extract facts as N-ary tuples using dependency parsing [102]. A year later, ClausIE [103] was developed, which identified clauses in an English sentence and then extracted facts by classifying the identified clauses using rules. In order to identify the relations or entities, dependency parsing of sentences was a crucial step in ClausIE and many other works [41, 42, 30, 43]. Built as an improvement to ClausIE, the MinIE [104] tool generated much more fine-grained and concise facts as compared to ClausIE. The triples generated by MinIE had dictionary-like attributes containing the information about certainty, polarity, and knowledge source. Due to the availability of manually annotated data for the English, much of the recent OIE research is based on deep neural architectures where the triple extraction problem is divided into the following two sub-problems: (a) Relation extraction and (b) Argument (*head*/*tail*) extraction using features from the extracted relation [105]. Span selection (using sequence labeling paradigm) is a common practice to extract relations and their corresponding arguments in such OIE methods [106].

The development of OIE tools for languages other than English is impeded by the lack of annotated resources available for them. Although, there are some notable initiatives in domain-specific IE for Hindi [107]. To the best of our knowledge, there has not been any work done yet, exclusively for OIE in Indic languages. However, the field of language-independent (multilingual)

OIE started in 2015 with two methods. The first method was developed by Manaal and Kumar (*M&K*) [108], where the authors translated the source language to English using Google translate and then extracted triples using the OLLIE tool. The English triples were projected back to their source language through word alignments. It could handle as many languages as Google can translate, but machine translation has not been regarded as a sustainable solution for OIE due to translation errors [33]. The second method was a rule-based triple extractor called *ArgOE* [43]. In order to generate triples, it expects dependency parse of a sentence in CoNLL format as input. However, the extracted triples contain only verb-mediated relations. *PredPatt* [41] was developed a year later, which also relied on a dependency parse tree and hand-crafted rules to identify predicate-argument structure in a sentence. Another work called *Multi2OIE* modeled the problem of identifying predicate-argument structure through two sequence-labeling tasks using mBERT embeddings and multi-head attention blocks [105]. The first task identified all the predicates in a sentence, and the second task identified all arguments associated with each predicate. One limitation of identifying predicate-argument structure is the post-processing that is needed to extract triples (or relations) out of them.

2.3 Hate-speech Detection

Employing machine learning (ML) algorithms has been the most common method to automatically detect hate-speech in text [109, 110]. Training traditional models (like SVM, decision trees, etc) using the extracted features (like N-grams, POS tags, word embeddings, etc) from text was a general approach in many of the earlier works [111, 112, 113]. Dinakar et al. [114] used world knowledge through the ontologies in ConceptNet to detect hostility for members of LGBT community in online forums.

Due to availability of greater number of annotated data, hate-speech detection methods based on deep learning (CNN/LSTM/GRU) started outperforming traditional ML models [115, 116]. Zampieri et al. [117] made an observation that in 2019 Semeval task on Offensive Language Identification majority of participants used a deep learning based method. However, Kwok and Wang [118] has previously observed that predictions of hate-speech detection methods are largely influenced by the presence of some specific offensive words. In 2018, attention-based BERT model achieved state-of-art performance on various downstream tasks such as Question-Answering and Named Entity Recognition [60]. In the subsequent works of hate-speech detection, it was observed that fine-tuning the pretrained model on the annotated data for hate-speech detection yielded better results as compared to other methods[119, 120, 121, 122].

We observed that most of the research in hate-speech detection was lim-

ited to English. Though there are some notable works for other European languages like Dutch [123], German [124], and Italian [125], hate-speech detection in Indian languages has not been explored to the same extend as English. However, with the advent of multilingual pretrained transformer models (like multilingual BERT), many of the recent works dealing with hate-speech detection in Indian languages have been focused on fine-tuning transformers [126, 127]. Velankar et al. [128] has demonstrated that fine-tuning transformer models on hate-speech detection performs better than rigorous hyperparameter tuning of deep learning methods.

2.4 Coreference Resolution

Hobbs [129] employed constituency tree parsing and hand-crafted rules to resolve coreferences. Lappin and Leass [130] used salience based features and mention-pair architecture for the same. Other methods that were used for coreference resolution involves centering theory [131] or knowledge-rich approaches with hand-crafted rules [132]. Syntax-based features have been popular method in earlier works of anaphora resolution [133, 134].

The recent best-performing models for coreference resolution have been based on neural mention-ranking architecture [135, 136]. Joshi et al. [137] pretrained a transformer model using span-based objective instead of word-level, and observed the best performance on coreference resolution task in English. However, anaphora resolution in Indian languages (specifically Hindi) has limited to theoretical approaches [138], models without mention-detection pipeline [139], and approaches that rely on morphological features [140, 141].

2.5 Research Gaps Identified

Our extensive literature survey has made us aware about various areas that are quite under explored in the field of indic-NLP. We formulate the identified research gaps as follows:

1. Considering the under-developed maternal health infrastructure in India. There is a need to propose automated solutions to provide healthcare related information to an end-user in remote places of India.

2. The field of Open Information Extraction (OIE) from unstructured text in Indic languages has not been explored much. Moreover, the effectiveness of existing multilingual OIE techniques has to be evaluated on Indic languages.

3. There is a scarcity of annotated resources for automatic evaluation of automatically generated triples for Indic languages.

4. Incorporating dependency structure of a sentence in generating dependency-aware contextual embeddings is an under-explored field in Indic-NLP.

5. A publicly available tool for coreference resolution in Indic languages is required to be built using the state-of-art coreference resolution techniques.

6. Pretraining of transformer models using different subword tokenization methods and pretraining tasks are to be investigated extensively.

7. Construction of knowledge-graph using extracted triples from unstructured text in Indian languages needs to be explored as well.

Chapter 3

Objectives

Considering the identified research gaps, we determine to achieve the following objectives:

1. Build a question-answering system (chatbot) for maternal workers using different sentence similarity approaches.
2. Develop a triple extractor from unstructured text in Indian languages, and explore their applicability in automatically answering questions from long documents.
3. In order to improve the chunker for triple extraction, we aim to transform subword-level transformer embeddings into Dependency-aware word-level Transformer (DaT) embeddings.
4. Perform automatic pronoun resolution in an end-to-end manner on a paragraph in Indian language. It will be used as a preprocessing step for the triple extraction process.
5. Implement long context question answering using various components from the above objectives
6. Pretrain a transformer model using a different pretraining objective instead of Masked Language Modelling (MLM), and use it to develop a QnA system for Indic languages.

3.1 Mission

In order to develop a QnA framework for Indic languages, we aim to build various NLP tools, and explore different paradigms for better contextual embeddings of Indic text. Our preliminary literature survey highlights a scarcity of such work that deals with Long-context Question Answering in Indic languages. Hence, it is our conviction that the proposed work is a critical step in the direction of advancement of research in indic-NLP.

Chapter 4

Progress Till Now

4.1 FAQ retrieval based Hindi Chatbot for healthcare workers

We aimed to contribute to the understanding of the performance of different algorithmic approaches in reducing workload of healthcare professionals by means of QnA support. Figure 4.1 illustrates the overall architecture of the proposed chatbot. Due to the COVID-19 pandemic, on field testing of AI models was not possible, therefore in order to evaluate the models on real-time data, a total of 336 new user queries (q) were collected from healthcare workers with the help of a non-governmental organization (NGO) partner. We annotated these 336 queries with relevant questions (Q) from the FAQ database. For each query q, authors identified completely matching, and partially matching QA pairs from ASHA-FAQ database. In our work, both

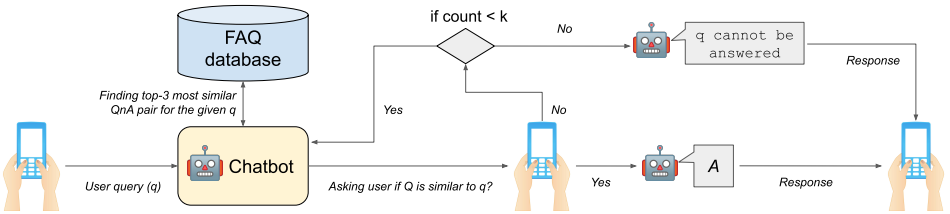


Figure 4.1: Architecture of the proposed chatbot. A user query (q) can be in Devanagari script or in Roman script. Using the best sentence similarity approach, the chatbot fetches top-k most similar Question-Answer (QnA) pairs from the FAQ database and stores them in a queue. User feedback is taken for one question (Q) at a time. If Q at rank 1 is similar to q then its corresponding Answer (A) is sent to the user as response, otherwise Q at rank 2 is sent to the user for the feedback. Among the k suggestions, if the user found not a single Q similar to q, then the chatbot responds *'q cannot be answered'*.

	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 4.1: Comparison of all three primary approaches on *hold-out test set* for top-3 suggestions. Ensemble ($\mathcal{E}$) is obtained by taking the best performing models, highlighted with yellow, from each of the primary approach.

kind of matching (complete and partial) had been treated as relevant. It has been found that, among 336 queries, only 270 user queries had at least one relevant question in the database. These 270 questions will be treated as the *hold-out test set* to asses our AI models.

We have experimented with different approaches to extract the most relevant question-answer (QA) pairs from a FAQ database. The three primary approaches used in this work are (i) Dependency Tree Pruning (DTP) method, (ii) Cosine Similarity (COS) method, and (ii) Sentence-pair Paraphrasing Classifier (SPC). We tackled the polysemous words in healthcare-related queries (like *sugar* vs *diabetes*), by maintaining buckets of such words. If a single word from a bucket is encountered in either q or Q_i, then rest of the words from the bucket are added to the text. Expanding the query in such a manner is called *query-expansion (q-e)* in the domain of automatic question answering [93]. Table 4.1 shows the performance boost in DTP and COS approach due to *q-e* variation. It was also observed that an ensemble ($\mathcal{E}$) model of all three primary approaches was found to be performing better than other methods.

Given a GPU-enabled server, our proposed chatbot can be freely deployed on instant messaging services like Telegram[1]. We also establish a relationship between the diversity of questions in FAQ-database, with the performance of our chatbot. We created a pruned *hold-out test-set* by removing the queries which had 1 or less relevant questions in FAQ database. The ensemble model evaluated on this pruned dataset is denoted by $\mathcal{E}^{p1}$. Number of user queries in the pruned set is written in the subscript. For example: removing user queries that have 3 or less relevant questions in FAQ database would result in *p3* set. We observed that *p3* set contains 105 user queries, each having 4 or more relevant questions in the FAQ database. The ensemble model evaluated on this *p3* set would be denoted by $\mathcal{E}^{p3}_{\#105}$. Figure 4.2 portrays that the performance of our best method ($\mathcal{E}$) improves when there are large number of user queries in the test set (> 100), and each user query has a large number of relevant questions in FAQ database (> 3).

[1] `telegram.org`

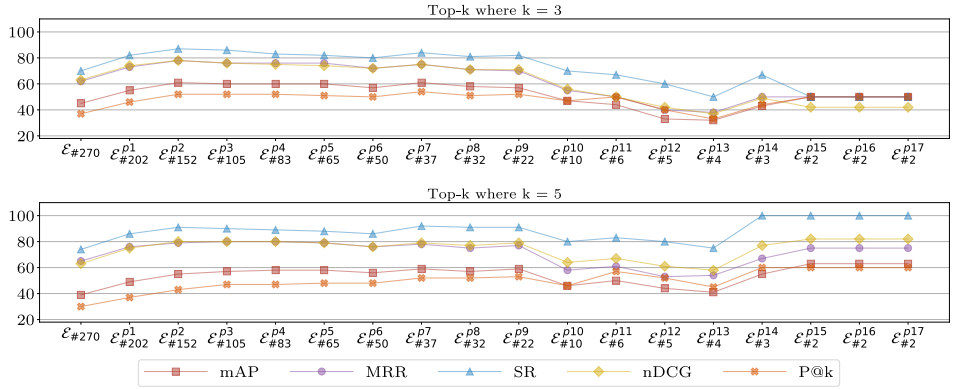


Figure 4.2: Performance of ensemble models on test-sets with different level of pruning. The label $\mathcal{E}^{p2}_{\#152}$ represents the performance of ensemble model on a *hold-out test-set* that is pruned by removing all user queries that have 2 or less relevant questions in the FAQ database. The resulting pruned test-set (*p2* set) has 152 user queries.

4.2 Triple Extractor for Indian Languages

Our work is primarily focused on automatically extracting triples from Hindi sentences since all the authors of this work are familiar with the language. However, we have discussed the performance of our method on other low-resource Indic languages as well. The main contributions of our work are as follows: (a) We release a transformer model which is fine-tuned on manually annotated chunks from six natural languages (*Hindi*, *English*, *Urdu*, *Nepali*, *Gujarati*, and *Bengali*). The resulting model is able to perform chunking on languages that it has not seen during the fine-tuning phase. (b) We propose a greedy algorithm to extract triples for Hindi sentences. Figure 4.3 describes the overall procedure of generating triples from unstructured text. (c) We create and release an OIE benchmark dataset for Hindi sentences, *Hindi-BenchIE*, to facilitate the automatic evaluation of machine-generated triples.

Since the task of chunking is defined as a sequence labeling task i.e. one prediction for each word, we needed word-level contextual embeddings for a given sentence. In this work, we tried a different approaches. Architecture of the proposed chunking model is illustrated in Figure 4.4. The effectiveness of our approach is empirically validated in Table 4.2.

Compared to CRF, our chunker gave better accuracy on the languages that it had never seen during the learning or fine-tuning phase. Table 4.3 compares our fine-tuned XLM chunker and the sklearn CRF model. As shown in Table 4.4, our method generated meaningful triples (containing all the information of the sentence) for the maximum number of sentences as compared to other baselines. Moreover, our method performs better than

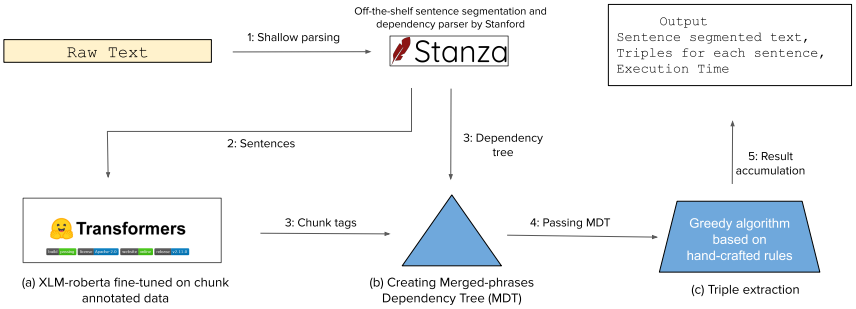


Figure 4.3: Overall architecture of the *IndIE* tool. The three primary steps are (a) Chunk tag prediction, (b) Creating Merged-phrases Dependency Tree (MDT), and (c) Triple generation. The three steps are run for each sentence segmented by the Stanza library [142].

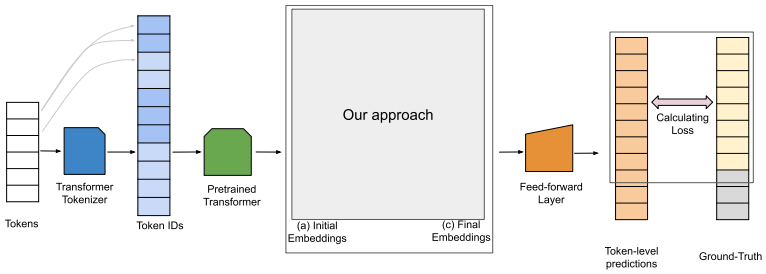


Figure 4.4: Architecture of the proposed chunking model. Word token IDs are highlighted with blue, whereas sub-word token IDs are highlighted in dark blue. Our approach takes the initial embeddings and transforms them. The resultant embeddings are zero-padded, highlighted with grey color, in order to produce the final embeddings in step (c). The padded tokens are ignored while calculating the loss for back-propagation.

Classification Layers	Common approach in literature	Another approach in practice	Our approach
1	82±10 (50±20)	89±0.5 (62±1.0)	**91±0.0** (**65±0.5**)
2	86±1.8 (51±6.2)	89±0.5 (54±7.4)	**90±0.5** (**54±4.5**)
3	79±14 (43±13)	82±11 (41±12)	**90±0.5** (**48±2.2**)

Table 4.2: Accuracy (macro average) of three approaches for solving the sub-word token embeddings for chunking task on four different random seeds.

other baselines on automatic evaluation of triples. Table 4.5 shows the performance comparison of different OIE methods.

Model	Hindi	English	Urdu	Nepali	Gujarati	Bengali
XLM	**78%**	**60%**	**84%**	**65%**	**56%**	**66%**
CRF	67%	56%	71%	58%	53%	53%

Table 4.3: A comparison of (fine-tuned) XLM chunker and CRF chunker on the languages which are removed from training-set. The numbers represent the accuracy obtained by each model when sentences from the given language are used only in the test-set.

Image options	*ArgOE*	*M&K*	*Multi2OIE*	*PredPatt*	*IndIE*
No information	17%	28%	**71%**	5%	4%
Most information	22%	**44%**	24%	29%	17%
All information	61%	28%	5%	66%	**79%**

Table 4.4: Percentage of sentences having no/most/all information in the image representation of their generated triples. The method which generates maximum triples with '*All information*' is considered the best method.

4.3 DaT embeddings for hate-speech detection

In the initial experiments, we have observed that using the flattened token embeddings gives better accuracy than using CLS embedding or pooler output embedding alone. Moreover, using flattened word-level contextual embeddings gives better accuracy than flattened subword-level embeddings. We have observed that the best approach to prepare word-level embeddings is to take average of subword token embeddings.

We observed that hate-speech detection using DaT embeddings gives better mean accuracy than hape-speech detection without DaT embeddings over multiple random seeds. We also implemented LRP technique for explaining the predictions of our hate-speech detector. As shown in Figure 4.5, we observed that the LRP technique we implemented gave more intuitive explanations behind the hate-speech prediction as compared to other methods.

	ArgOE	*M&K*	*Multi2OIE*	*PredPatt*	*IndIE*
Precision	0.26	0.14	0.12	0.37	**0.62**
Recall	0.06	0.16	0.03	0.08	**0.62**
F1-score	0.10	0.15	0.05	0.14	**0.62**

Table 4.5: Performance of different OIE methods on *Hindi-BenchIE* golden set. It is observed that our method (*IndIE*) outperforms other methods on the automatic evaluation benchmark.

Figure 4.5: The following three methods used for explaining the predictions of the trained hate-speech classifier: LRP (top), LIME (mid), and SHAP (below).

Chapter 5

Future Plans

We intend to evaluate the effectiveness of DaT embeddings for other tasks such as chunking. However, considering the progress at this point, the broader aim is to develop models for different downstream applications and explore different pretraining paradigms for transformers in order to build long-context QnA system for Indic languages. We intend to explore various methods to improve the contextual embeddings produced by transformers. We also intend to develop models for Indic languages that could solve various supporting tasks (such as pronoun resolution) which would be helpful as a preprocessing step for the previous methods.

5.1 Pronoun resolution

We plan to develop an end-to-end model for pronomial resolution in Hindi text. It has been shown that it is better to jointly learn anaphoricity detection and coreference together with a single loss [76]. Hence, the intended model will be trained using multi-task learning. Figure 5.1 illustrated the architecture of the proposed model.

The proposed model will be an extension to the work of Lee et al. [135] with a novel direction of multi-task learning on Hindi sentences. We intend to fine-tune the proposed model on the WikiCoref dataset [143], and the dataset by Mujadia et al. [144]. The former is a coref annotated dataset in English whereas the later is in Hindi. We have selected these two datasets because to the best of our knowledge these are the only datasets which have annotated pronomial coreference as a separate type of coreference.

5.2 Indic-SpanBERT

It has been observed that using a span-based pretraining task for transformer encoders results in improved performance on various downstream tasks [137]. The current model has been pretrained on English text dump only. We

intend to use our multilingual chunker and predict chunk-based spans as the pretraining task.

5.3 Long-context Question Answering

Widely used datasets for automatic Question Answering (QnA) includes datasets like SQuAD [145] which has a short-context for each question. Most of the QnA systems for Indic languages are also built for short-context paragraphs. We intend to explore the possibility of using OIE for answering long-context questions. As a baseline, we plan to use a combination of Answer-Sentence Selection (AS2) model and Machine Comprehension (MC) model for Indic languages. The XQA dataset proposed by Liu et al. [146] will be used as the evaluation dataset. It is to be noted that long-context question answering can either be open-domain question answering (like XQA) or domain-specific question answering (like healthcare chatbot). Since we haven't come across an annotated domain-specific dataset for long-context question answering, we intend to direct our efforts for open-domain question answering system.

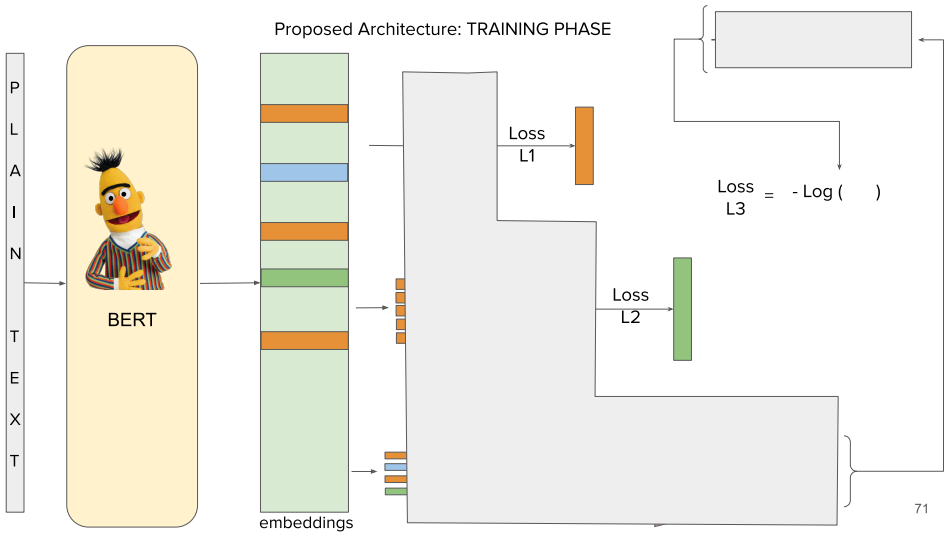


Figure 5.1: Training phase of the proposed end-to-end neural architecture for pronomial resolution in Hindi. The horizontal bars in orange, blue, and green represents *mention* words. Pronomial word is represented by green horizontal bar, whereas its gold antecedent is represented by a blue horizontal bar.

5.4 Timeline

After considering the progress of work and plans for future, the expected timeline for making a significant contribution in the field of indic-NLP and completing the requirements for the doctoral program is as follows:

1. Aug 2022 - Oct 2022
    - Incorporate the internal reviews for maternal chatbot, and submit the manuscript to a journal like ACM Health (average citation per article is 2) or IEEE Transactions on CSS (Impact score is 5).
    - Finish the experimentation pipeline for DaT embeddings. Prepare the first draft.
    - Data preprocessing for pronoun resolution.
2. Nov 2022 - Jan 2023
    - Incorporate internal reviews for DaT embeddings paper, and submit the draft to ACL Rolling Reviews (ARR). Target AACL.
    - Build the multi-task learning pipeline for pronoun resolution.
    - Prepare data and dataloader for Indic-SpanBERT pretraining. Explore the usability of bert and distilbert.
3. Feb 2023 - Mar 2023
    - Implement the baselines for pronoun resolution and design evaluation metrics.
    - Explore the usability of OIE in long-context question answering.
    - Use a combination of Answer Sentence Selection (AS2) and Machine Comprehension (MC) to solve long-context question answering.
    - Evaluate Indic-SpanBERT on various downstream tasks in Indic languages. Document the work as a short paper.
4. May 2023 - Aug 2023
    - Prepare the draft for pronoun resolution, and long-context question answering. Collect internal reviews. Modify and submit in EMNLP.
    - Begin a survey on work done in Indic NLP.
5. Sep 2023 - Dec 2023
    - Begin thesis writing.
    - Draft review cycles.
    - Defense[1].

[1] After 4.5 years into the PhD program

Chapter 6

Publications

1. Mishra, Ritwik, Simranjeet Singh, Rajiv Ratn Shah, Ponnurangam Kumaraguru, and Pushpak Bhattacharya. "IndIE: A Multilingual Open Information Extraction Tool For Indic Languages". 2022. **[Submitted to a special issue of TALLIP]**

2. Mishra, Ritwik, Simranjeet Singh, Jasmeet Kaur, Rajiv Ratn Shah, and Pushpendra Singh. "Exploring the Use of Chatbots for Supporting Maternal and Child Health in Resource-constrained Environments". 2022. **[Draft ready. Under internal review]**

6.1 Miscellaneous

1. Mishra, Ritwik, Ponnurangam Kumaraguru, Rajiv Ratn Shah, Aanshul Sadaria, Shashank Srikanth, Kanay Gupta, Himanshu Bhatia, and Pratik Jain. "Analyzing traffic violations through e-challan system in metropolitan cities (workshop paper)." In 2020 IEEE Sixth International Conference on Multimedia Big Data (BigMM), pp. 485-493. IEEE, 2020.

Acknowledgments

1. I would like to express my gratitude to pillars group members for their valuable guidance.

2. I would like to thank Simranjeet, Samarth, Ajeet, and Jasmeet for being diligent co-authors.

3. I would also like to thank Prof Pushpak Bhattacharyya, and CFILT lab members for providing me great insights and hosting me at IIT Bombay under Anveshan Setu program.

4. I would also like to University Grant Commission (UGC) Junior Research Fellowship (JRF) / Senior Research Fellowship (SRF) for funding my PhD program

Bibliography

[1] Mika Westerlund. The emergence of deepfake technology: A review. *Technology Innovation Management Review*, 9(11), 2019.

[2] Yaniv Leviathan and Yossi Matias. Google duplex: an ai system for accomplishing real-world tasks over the phone. 2018.

[3] Yuval N. Harari, David Vandermeulen, and Daniel Casanave. *Sapiens: Chapter 7*. Albin Michel, 2020.

[4] Murray B Emeneau. India as a lingustic area. *Language*, 32(1):3–16, 1956.

[5] What are the top 200 most spoken languages?, Feb 2021.

[6] Census of india, 2011.

[7] KPMG and Google. Indian languages - defining india's internet. Apr 2017.

[8] Julia Hirschberg and Christopher D Manning. Advances in natural language processing. *Science*, 349(6245):261–266, 2015.

[9] Bert F Green Jr, Alice K Wolf, Carol Chomsky, and Kenneth Laughery. Baseball: an automatic question-answerer. In *Papers presented at the May 9-11, 1961, western joint IRE-AIEE-ACM computer conference*, pages 219–224, 1961.

[10] Nicolas Pfeuffer, Alexander Benlian, Henner Gimpel, and Oliver Hinz. Anthropomorphic information systems. *Business & Information Systems Engineering*, 61(4):523–533, 2019.

[11] Surbhi Khurana and Amita Dev. Present scenario of emotionally intelligent voice-based conversational agents in india. In *Artificial Intelligence and Speech Technology*, pages 63–75. CRC Press, 2021.

[12] Liliana Laranjo, Adam G Dunn, Huong Ly Tong, Ahmet Baki Kocaballi, Jessica Chen, Rabia Bashir, Didi Surian, Blanca Gallego, Farah Magrabi, Annie YS Lau, et al. Conversational agents in healthcare: a

systematic review. *Journal of the American Medical Informatics Association*, 25(9):1248–1258, 2018.

[13] Joao Luis Zeni Montenegro, Cristiano André da Costa, and Rodrigo da Rosa Righi. Survey of conversational agents in health. *Expert Systems with Applications*, 129:56–67, 2019.

[14] Debasis Barik and Amit Thorat. Issues of unequal access to public health in india. *Frontiers in public health*, 3:245, 2015.

[15] Shalini Kanuganti, Ashoke Kumar Sarkar, Ajit Pratap Singh, and Shriniwas S Arkatkar. Quantification of accessibility to health facilities in rural areas. *Case Studies on Transport Policy*, 3(3):311–320, 2015.

[16] Étienne V Langlois, Malgorzata Miszkurka, Maria Victoria Zunzunegui, Abdul Ghaffar, Daniela Ziegler, and Igor Karp. Inequities in postnatal care in low-and middle-income countries: a systematic review and meta-analysis. *Bulletin of the World Health Organization*, 93:259–270G, 2015.

[17] Ashavaree Das and Madhurima Sarkar. 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, 2014.

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

[19] Shafquat Hussain, Omid Ameri Sianaki, and Nedal Ababneh. A survey on conversational agents/chatbots classification and design techniques. In *Workshops of the International Conference on Advanced Information Networking and Applications*, pages 946–956. Springer, 2019.

[20] Wataru Sakata, Tomohide Shibata, Ribeka Tanaka, and Sadao Kurohashi. 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, 2019.

[21] Mladen Karan, Lovro Žmak, and Jan Šnajder. Frequently asked questions retrieval for croatian based on semantic textual similarity. In *Proceedings of the 4th Biennial International Workshop on Balto-Slavic Natural Language Processing*, pages 24–33, 2013.

[22] Liang Zhang, Yan Yang, Jie Zhou, Chengcai Chen, and Liang He. Retrieval-polished response generation for chatbot. *IEEE Access*, 8:123882–123890, 2020.

[23] Yu Wu, Zhoujun Li, Wei Wu, and Ming Zhou. Response selection with topic clues for retrieval-based chatbots. *Neurocomputing*, 316:251–261, 2018.

[24] Chongyang Tao, Wei Wu, Can Xu, Wenpeng Hu, Dongyan Zhao, and Rui Yan. Multi-representation fusion network for multi-turn response selection in retrieval-based chatbots. In *Proceedings of the twelfth ACM international conference on web search and data mining*, pages 267–275, 2019.

[25] Pranav Bhagat, Sachin Kumar Prajapati, and Aaditeshwar Seth. 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, 2020. Association for Computing Machinery.

[26] Iz Beltagy, Matthew E Peters, and Arman Cohan. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*, 2020.

[27] Pradeep Dasigi, Kyle Lo, Iz Beltagy, Arman Cohan, Noah A. Smith, and Matt Gardner. A dataset of information-seeking questions and answers anchored in research papers. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4599–4610, Online, June 2021. Association for Computational Linguistics.

[28] YORICK WILKS. Text searching with templates. *Cambridge Language Research Unit Memo, ML*, (165), 1964.

[29] Yorick Wilks. Information Extraction as a core language technology: What is IE? In *of Lecture Notes in Computer Science, chapter In M-T. Pazienza (ed.), Information Extraction*, pages 14–18. Springer, 1997.

[30] Pablo Gamallo, Marcos Garcia, and Santiago Fernández-Lanza. Dependency-based open information extraction. In *Proceedings of the joint workshop on unsupervised and semi-supervised learning in NLP*, pages 10–18, 2012.

[31] Iqra Muhammad, Anna Kearney, Carrol Gamble, Frans Coenen, and Paula Williamson. Open information extraction for knowledge graph construction. In *International Conference on Database and Expert Systems Applications*, pages 103–113. Springer, 2020.

[32] Xueling Lin, Haoyang Li, Hao Xin, Zijian Li, and Lei Chen. Kbpearl: a knowledge base population system supported by joint entity and relation linking. *Proceedings of the VLDB Endowment*, 13(7):1035–1049, 2020.

[33] Daniela Barreiro Claro, Marlo Souza, Clarissa Castellã Xavier, and Leandro Oliveira. Multilingual open information extraction: Challenges and opportunities. *Information*, 10(7):228, 2019.

[34] Zhao Yan, Duyu Tang, Nan Duan, Shujie Liu, Wendi Wang, Daxin Jiang, Ming Zhou, and Zhoujun Li. Assertion-based qa with question-aware open information extraction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[35] Yumo Xu and Mirella Lapata. Abstractive query focused summarization with query-free resources. *CoRR*, abs/2012.14774, 2020.

[36] Giovanni Siragusa, Rohan Nanda, Valeria De Paiva, and Luigi Di Caro. Relating legal entities via open information extraction. In *Research Conference on Metadata and Semantics Research*, pages 181–187. Springer, 2018.

[37] Tara Mohanan. Case ocp: A constraint on word order in hindi. *Theoretical perspectives on word order in South Asian languages*, 185:216, 1994.

[38] Jaya S Nagendra. Basic grammar, hindi. In *A Brief History of Languages*, volume 1, page 190. Atlantic, 2019.

[39] Kiril Gashteovski, Mingying Yu, Bhushan Kotnis, Carolin Lawrence, Goran Glavas, and Mathias Niepert. Benchie: Open information extraction evaluation based on facts, not tokens. *arXiv preprint arXiv:2109.06850*, 2021.

[40] Sangnie Bhardwaj, Samarth Aggarwal, and Mausam Mausam. CaRB: A crowdsourced benchmark for open IE. In *Proceedings of the 2019 Conference on EMNLP-IJCNLP*, pages 6262–6267, Hong Kong, China, November 2019. Association for Computational Linguistics.

[41] Aaron Steven White, Drew Reisinger, Keisuke Sakaguchi, Tim Vieira, Sheng Zhang, Rachel Rudinger, Kyle Rawlins, and Benjamin Van Durme. Universal decompositional semantics on universal dependencies. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1713–1723, 2016.

[42] Sheng Zhang, Xutai Ma, Rachel Rudinger, Kevin Duh, and Benjamin Van Durme. Cross-lingual decompositional semantic parsing. In *Pro-*

ceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 1664–1675, 2018.

[43] Pablo Gamallo and Marcos Garcia. Multilingual open information extraction. In *Portuguese Conference on Artificial Intelligence*, pages 711–722. Springer, 2015.

[44] John Burton-Page. Compound and conjunct verbs in hindi1. *Bulletin of the School of Oriental and African Studies*, 19(3):469–478, 1957.

[45] Shamim Fatma. Conjunct verbs in hindi. *Trends in Hindi Linguistics*, pages 217–244, 2018.

[46] Thi Oanh Tran, Phuong Le Hong, et al. Improving sequence tagging for vietnamese text using transformer-based neural models. In *Proceedings of the 34th Pacific Asia Conference on Language, Information and Computation*, pages 13–20, 2020.

[47] Ehsan Doostmohammadi, Minoo Nassajian, and Adel Rahimi. Persian ezafe recognition using transformers and its role in part-of-speech tagging. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 961–971, 2020.

[48] Hongwei Li, Hongyan Mao, and Jingzi Wang. Part-of-speech tagging with rule-based data preprocessing and transformer. *Electronics*, 11(1):56, 2021.

[49] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *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)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[50] Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Édouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, 2020.

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

[52] Taeuk Kim, Jihun Choi, Daniel Edmiston, and Sang-goo Lee. Are pre-trained language models aware of phrases? simple but strong baselines for grammar induction. *CoRR*, abs/2002.00737, 2020.

[53] David Vilares, Michalina Strzyz, Anders Søgaard, and Carlos Gómez-Rodríguez. Parsing as pretraining. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 9114–9121, 2020.

[54] Allyson Ettinger. What BERT is not: Lessons from a new suite of psycholinguistic diagnostics for language models. *CoRR*, abs/1907.13528, 2019.

[55] Zhanming Jie, Aldrian Obaja Muis, and Wei Lu. Efficient dependency-guided named entity recognition. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[56] Diego Marcheggiani and Ivan Titov. Encoding sentences with graph convolutional networks for semantic role labeling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1506–1515, Copenhagen, Denmark, September 2017. Association for Computational Linguistics.

[57] Yuhao Zhang, Peng Qi, and Christopher D. Manning. Graph convolution over pruned dependency trees improves relation extraction. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2205–2215, Brussels, Belgium, October-November 2018. Association for Computational Linguistics.

[58] Tao Ji, Yuanbin Wu, and Man Lan. Graph-based dependency parsing with graph neural networks. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2475–2485, 2019.

[59] Max Welling and Thomas N Kipf. Semi-supervised classification with graph convolutional networks. In *J. International Conference on Learning Representations (ICLR 2017)*, 2016.

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

[61] Shiladitya Bhattacharya, Siddharth Singh, Ritesh Kumar, Akanksha Bansal, Akash Bhagat, Yogesh Dawer, Bornini Lahiri, and Atul Kr Ojha. Developing a multilingual annotated corpus of misogyny and aggression. In *Proceedings of the Second Workshop on Trolling, Aggression and Cyberbullying*, pages 158–168, 2020.

[62] Puneet Mathur, Ramit Sawhney, Meghna Ayyar, and Rajiv Shah. Did you offend me? classification of offensive tweets in Hinglish language. In *Proceedings of the 2nd Workshop on Abusive Language Online (ALW2)*, pages 138–148, Brussels, Belgium, October 2018. Association for Computational Linguistics.

[63] Hammad Rizwan, Muhammad Haroon Shakeel, and Asim Karim. Hate-speech and offensive language detection in roman urdu. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2512–2522, 2020.

[64] Parth Patwa, Mohit Bhardwaj, Vineeth Guptha, Gitanjali Kumari, Shivam Sharma, Srinivas PYKL, Amitava Das, Asif Ekbal, Shad Akhtar, and Tanmoy Chakraborty. Overview of constraint 2021 shared tasks: Detecting english covid-19 fake news and hindi hostile posts. In *Proceedings of the First Workshop on Combating Online Hostile Posts in Regional Languages during Emergency Situation (CONSTRAINT)*. Springer, 2021.

[65] Grégoire Montavon, Alexander Binder, Sebastian Lapuschkin, Wojciech Samek, and Klaus-Robert Müller. Layer-wise relevance propagation: an overview. *Explainable AI: interpreting, explaining and visualizing deep learning*, pages 193–209, 2019.

[66] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144, 2016.

[67] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 4765–4774. Curran Associates, Inc., 2017.

[68] Dimitris Papadopoulos, Nikolaos Papadakis, and Antonis Litke. A methodology for open information extraction and representation from large scientific corpora: the cord-19 data exploration use case. *Applied Sciences*, 10(16):5630, 2020.

[69] Dimitris Papadopoulos, Nikolaos Papadakis, and Nikolaos Matsatsinis. Penelopie: Enabling open information extraction for the greek language through machine translation. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Student Research Workshop*, pages 23–29, 2021.

[70] Kiril Gashteovski, Sebastian Wanner, Sven Hertling, Samuel Broscheit, and Rainer Gemulla. Opiec: An open information extraction corpus. In *Automated Knowledge Base Construction (AKBC)*, 2018.

[71] Ellery Smith, Dimitris Papadopoulos, Martin Braschler, and Kurt Stockinger. Lillie: Information extraction and database integration using linguistics and learning-based algorithms. *Information Systems*, 105:101938, 2022.

[72] Daniel Jurafsky and James H Martin. *Speech and language processing (draft)*. 2022. Available from: `https://web.stanford.edu/~jurafsky/slp3/`.

[73] Jing Lu and Vincent Ng. Event coreference resolution: a survey of two decades of research. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pages 5479–5486, 2018.

[74] Rhea Sukthanker, Soujanya Poria, Erik Cambria, and Ramkumar Thirunavukarasu. Anaphora and coreference resolution: A review. *Information Fusion*, 59:139–162, 2020.

[75] Vincent Ng. Supervised noun phrase coreference research: The first fifteen years. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1396–1411, Uppsala, Sweden, July 2010. Association for Computational Linguistics.

[76] Altaf Rahman and Vincent Ng. Supervised models for coreference resolution. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 968–977, Singapore, August 2009. Association for Computational Linguistics.

[77] Sebastian Martschat and Michael Strube. Latent structures for coreference resolution. *Transactions of the Association for Computational Linguistics*, 3:405–418, 2015.

[78] Bonnie Lynn Webber. Discourse deixis: reference to discourse segments. In *Proceedings of the 26th annual meeting on Association for Computational Linguistics*, pages 113–122, 1988.

[79] Ning Yang, Jingyu Zhang, Lijun Ma, and Zhi Lu. A study of zero anaphora resolution in chinese discourse: From the perspective of psycholinguistics. *Frontiers in Psychology*, 12, 2021.

[80] Soomin Kim, Jinsu Eun, Changhoon Oh, Bongwon Suh, and Joonhwan Lee. Bot in the bunch: Facilitating group chat discussion by improving efficiency and participation with a chatbot. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI '20,

page 1–13, New York, NY, USA, 2020. Association for Computing Machinery.

[81] Soomin Kim, Jinsu Eun, Joseph Seering, and Joonhwan Lee. Moderator chatbot for deliberative discussion: Effects of discussion structure and discussant facilitation. *Proc. ACM Hum.-Comput. Interact.*, 5(CSCW1), apr 2021.

[82] Yi-Chieh Lee, Naomi Yamashita, Yun Huang, and Wai Fu. "i hear you, i feel you": Encouraging deep self-disclosure through a chatbot. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI '20, page 1–12, New York, NY, USA, 2020. Association for Computing Machinery.

[83] Yuna Ahn, Yilin Zhang, Yujin Park, and Joonhwan Lee. A chatbot solution to chat app problems: Envisioning a chatbot counseling system for teenage victims of online sexual exploitation. In *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI EA '20, page 1–7, New York, NY, USA, 2020. Association for Computing Machinery.

[84] SangAh Park, Yoon Young Lee, Soobin Cho, Minjoon Kim, and Joongseek Lee. *"Knock Knock, Here Is an Answer from Next Door": Designing a Knowledge Sharing Chatbot to Connect Residents: Community Chatbot Design Case Study*, page 144–148. Association for Computing Machinery, New York, NY, USA, 2021.

[85] Niranjan Pai, Pradnya Supe, Shailesh Kore, Y. S. Nandanwar, Aparna Hegde, Edward Cutrell, and William Thies. Using automated voice calls to improve adherence to iron supplements during pregnancy: A pilot study. In *Proceedings of the Sixth International Conference on Information and Communication Technologies and Development: Full Papers - Volume 1*, ICTD '13, page 153–163, New York, NY, USA, 2013. Association for Computing Machinery.

[86] Konstantinos Kazakos, Siddhartha Asthana, Madeline Balaam, Mona Duggal, Amey Holden, Limalemla Jamir, Nanda Kishore Kannuri, Saurabh Kumar, Amarendar Reddy Manindla, Subhashini Arcot Manikam, GVS Murthy, Papreen Nahar, Peter Phillimore, Shreyaswi Sathyanath, Pushpendra Singh, Meenu Singh, Pete Wright, Deepika Yadav, and Patrick Olivier. A real-time ivr platform for community radio. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16, page 343–354, New York, NY, USA, 2016. Association for Computing Machinery.

[87] Eric Brill, Susan Dumais, and Michele Banko. 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, 2002.

[88] Govind Kothari, Sumit Negi, Tanveer A Faruquie, Venkatesan T Chakaravarthy, and L Venkata Subramaniam. 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, 2009.

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

[90] Khyathi Raghavi, Manoj Chinnakotla, and Manish Shrivastava. " answer ka type kya he? " learning to classify questions in code-mixed language. 05 2015.

[91] Khyathi Raghavi Chandu, Manoj Chinnakotla, Alan W Black, and Manish Shrivastava. Webshodh: A code mixed factoid question answering system for web. In *International Conference of the Cross-Language Evaluation Forum for European Languages*, pages 104–111. Springer, 2017.

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

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

[94] Jeanne E Daniel, Willie Brink, Ryan Eloff, and Charles Copley. 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, 2019.

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

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

[97] Rajat Mohanty, Anupama Dutta, and Pushpak Bhattacharyya. Semantically relatable sets: building blocks for representing semantics. In *MT Summit*, volume 5. Citeseer, 2005.

[98] Oren Etzioni, Michele Banko, Stephen Soderland, and Daniel S Weld. Open information extraction from the web. *Communications of the ACM*, 51(12):68–74, 2008.

[99] Janara Christensen, Stephen Soderland, and Oren Etzioni. An analysis of open information extraction based on semantic role labeling. In *Proceedings of the sixth international conference on Knowledge capture*, pages 113–120, 2011.

[100] Anthony Fader, Stephen Soderland, and Oren Etzioni. Identifying relations for open information extraction. In *Proceedings of the 2011 conference on empirical methods in natural language processing*, pages 1535–1545, 2011.

[101] Mausam, Michael Schmitz, Stephen Soderland, Robert Bart, and Oren Etzioni. Open language learning for information extraction. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 523–534, Jeju Island, Korea, July 2012. Association for Computational Linguistics.

[102] Alan Akbik and Alexander Löser. Kraken: N-ary facts in open information extraction. In *Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction (AKBC-WEKEX)*, pages 52–56, 2012.

[103] Luciano Del Corro and Rainer Gemulla. Clausie: clause-based open information extraction. In *Proceedings of the 22nd international conference on World Wide Web*, pages 355–366, 2013.

[104] Kiril Gashteovski, Rainer Gemulla, and Luciano del Corro. Minie: minimizing facts in open information extraction. Association for Computational Linguistics, 2017.

[105] Youngbin Ro, Yukyung Lee, and Pilsung Kang. Multi^{2}oie: Multilingual open information extraction based on multi-head attention with bert. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pages 1107–1117, 2020.

[106] Junlang Zhan and Hai Zhao. Span model for open information extraction on accurate corpus. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 9523–9530, 2020.

[107] Pattabhi RK Rao and Sobha Lalitha Devi. Eventxtract-il: Event extraction from newswires and social media text in indian languages@ fire 2018-an overview. In *FIRE (Working Notes)*, pages 282–290, 2018.

[108] Manaal Faruqui and Shankar Kumar. Multilingual open relation extraction using cross-lingual projection. In *North American Chapter of the ACL: Human Language Technologies*, pages 1351–1356, 2015.

[109] Paula Fortuna and Sérgio Nunes. A survey on automatic detection of hate speech in text. *ACM Computing Surveys (CSUR)*, 51(4):1–30, 2018.

[110] Anna Schmidt and Michael Wiegand. A survey on hate speech detection using natural language processing. In *Proceedings of the Fifth International Workshop on Natural Language Processing for Social Media, April 3, 2017, Valencia, Spain*, pages 1–10. Association for Computational Linguistics, 2019.

[111] Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. Automated hate speech detection and the problem of offensive language. In *Proceedings of the international AAAI conference on web and social media*, volume 11, pages 512–515, 2017.

[112] Edel Greevy and Alan F Smeaton. Classifying racist texts using a support vector machine. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 468–469, 2004.

[113] Pete Burnap and Matthew L Williams. Cyber hate speech on twitter: An application of machine classification and statistical modeling for policy and decision making. *Policy & internet*, 7(2):223–242, 2015.

[114] Karthik Dinakar, Birago Jones, Catherine Havasi, Henry Lieberman, and Rosalind Picard. Common sense reasoning for detection, prevention, and mitigation of cyberbullying. *ACM Transactions on Interactive Intelligent Systems (TiiS)*, 2(3):1–30, 2012.

[115] Son T Luu, Hung P Nguyen, Kiet Van Nguyen, and Ngan Luu-Thuy Nguyen. Comparison between traditional machine learning models and neural network models for vietnamese hate speech detection. In *2020 RIVF International Conference on Computing and Communication Technologies (RIVF)*, pages 1–6. IEEE, 2020.

[116] Pinkesh Badjatiya, Shashank Gupta, Manish Gupta, and Vasudeva Varma. Deep learning for hate speech detection in tweets. In *Proceedings of the 26th international conference on World Wide Web companion*, pages 759–760, 2017.

[117] Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. SemEval-2019 task 6: Identifying and categorizing offensive language in social media (OffensEval). In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 75–86, Minneapolis, Minnesota, USA, June 2019. Association for Computational Linguistics.

[118] Irene Kwok and Yuzhou Wang. Locate the hate: Detecting tweets against blacks. In *Twenty-seventh AAAI conference on artificial intelligence*, 2013.

[119] Marzieh Mozafari, Reza Farahbakhsh, and Noel Crespi. A bert-based transfer learning approach for hate speech detection in online social media. In *International Conference on Complex Networks and Their Applications*, pages 928–940. Springer, 2019.

[120] Sean MacAvaney, Hao-Ren Yao, Eugene Yang, Katina Russell, Nazli Goharian, and Ophir Frieder. Hate speech detection: Challenges and solutions. *PloS one*, 14(8):e0221152, 2019.

[121] Ping Liu, Wen Li, and Liang Zou. NULI at SemEval-2019 task 6: Transfer learning for offensive language detection using bidirectional transformers. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 87–91, Minneapolis, Minnesota, USA, June 2019. Association for Computational Linguistics.

[122] Wenjie Yin and Arkaitz Zubiaga. Towards generalisable hate speech detection: a review on obstacles and solutions. *PeerJ Computer Science*, 7:e598, 2021.

[123] Stéphan Tulkens, Lisa Hilte, Elise Lodewyckx, Ben Verhoeven, and Walter Daelemans. A dictionary-based approach to racism detection in dutch social media. In *Workshop Programme*, page 11, 2016.

[124] BjörnRoss MichaelRist GuillermoCarbonell and BenjaminCabrera NilsKurowsky MichaelWojatzki. Measuring the reliability of hate speech annotations: The case of the european refugee crisis. *Bochumer Linguistische Arbeitsberichte*, page 6, 2016.

[125] Fabio Del Vigna12, Andrea Cimino23, Felice Dell'Orletta, Marinella Petrocchi, and Maurizio Tesconi. Hate me, hate me not: Hate speech detection on facebook. In *Proceedings of the First Italian Conference on Cybersecurity (ITASEC17)*, pages 86–95, 2017.

[126] Thomas Mandl, Sandip Modha, Anand Kumar M, and Bharathi Raja Chakravarthi. Overview of the hasoc track at fire 2020: Hate speech

and offensive language identification in tamil, malayalam, hindi, english and german. In *Forum for information retrieval evaluation*, pages 29–32, 2020.

[127] Neeraj Vashistha and Arkaitz Zubiaga. Online multilingual hate speech detection: experimenting with hindi and english social media. *Information*, 12(1):5, 2020.

[128] Abhishek Velankar, Hrushikesh Patil, Amol Gore, Shubham Salunke, and Raviraj Joshi. Hate and offensive speech detection in hindi and marathi. *arXiv preprint arXiv:2110.12200*, 2021.

[129] Jerry R. Hobbs. Resolving pronoun references. *Lingua*, 44(4):311–338, 1978.

[130] Shalom Lappin and Herbert J Leass. An algorithm for pronominal anaphora resolution. *Computational linguistics*, 20(4):535–561, 1994.

[131] Susan E Brennan, Marilyn W Friedman, and Carl Pollard. A centering approach to pronouns. In *25th Annual Meeting of the Association for Computational Linguistics*, pages 155–162, 1987.

[132] T. Liang and D.S. Wu. Automatic pronominal anaphora resolution in english texts. *Computational Linguistics and Chinese Language Processing*, 9(1):21 – 40, 2004. Cited by: 23.

[133] Aria Haghighi and Dan Klein. Simple coreference resolution with rich syntactic and semantic features. In *Proceedings of the 2009 conference on empirical methods in natural language processing*, pages 1152–1161, 2009.

[134] Amir Zeldes and Shuo Zhang. When annotation schemes change rules help: A configurable approach to coreference resolution beyond OntoNotes. In *Proceedings of the Workshop on Coreference Resolution Beyond OntoNotes (CORBON 2016)*, pages 92–101, San Diego, California, June 2016. Association for Computational Linguistics.

[135] Kenton Lee, Luheng He, Mike Lewis, and Luke Zettlemoyer. End-to-end neural coreference resolution. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 188–197, Copenhagen, Denmark, September 2017. Association for Computational Linguistics.

[136] Kenton Lee, Luheng He, and Luke Zettlemoyer. Higher-order coreference resolution with coarse-to-fine inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2*

(Short Papers), pages 687–692, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[137] Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. SpanBERT: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77, 2020.

[138] Kamlesh Dutta, Nupur Prakash, and Saroj Kaushik. Resolving pronominal anaphora in hindi using hobbs algorithm. *Web Journal of Formal Computation and Cognitive Linguistics*, 1(10):5607–11, 2008.

[139] Praveen Dakwale, Vandan Mujadia, and Dipti Misra Sharma. A hybrid approach for anaphora resolution in hindi. In *Proceedings of the Sixth International Joint Conference on Natural Language Processing*, pages 977–981, 2013.

[140] Sobha Lalitha Devi, Vijay Sundar Ram, and Pattabhi RK Rao. A generic anaphora resolution engine for indian languages. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 1824–1833, 2014.

[141] Utpal Kumar Sikdar, Asif Ekbal, and Sriparna Saha. A generalized framework for anaphora resolution in indian languages. *Knowledge-Based Systems*, 109:147–159, 2016.

[142] Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 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*, 2020.

[143] Abbas Ghaddar and Philippe Langlais. Wikicoref: An english coreference-annotated corpus of wikipedia articles. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 136–142, 2016.

[144] Vandan Mujadia, Palash Gupta, and Dipti Misra Sharma. Coreference annotation scheme and relation types for hindi. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 161–168, 2016.

[145] Pranav Rajpurkar, Robin Jia, and Percy Liang. Know what you don't know: Unanswerable questions for squad. *arXiv preprint arXiv:1806.03822*, 2018.

[146] Jiahua Liu, Yankai Lin, Zhiyuan Liu, and Maosong Sun. Xqa: A cross-lingual open-domain question answering dataset. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2358–2368, 2019.