Kāraka-Based Answer Retrieval for Question Answering in Indic Languages

Devika Verma[*+], Ramprasad Joshi[*], Aiman Shivani[+], Rohan Gupta[+]
[*]BITS Pilani K K Birla Goa Campus
[+]Vishwakarma Institute of Information Technology
[*]{p20160010,rsj}@bits-pilani.ac.in [+]{aiman.21910033,rohan.21910456}@viit.ac.in

Abstract

*Kāraka*s from ancient Paninian grammar form a concise set of semantic roles that capture crucial aspect of sentence meaning pivoted on the action verb. In this paper, we propose employing a *kāraka*-based approach for retrieving answers in Indic question-answering systems. To study and evaluate this novel approach, empirical experiments are conducted over large benchmark corpora in Hindi and Marathi. The results obtained demonstrate the effectiveness of the proposed method. Additionally, we explore the varying impact of two approaches for extracting *kāraka*s. The literature surveyed and experiments conducted encourage hope that *kāraka* annotation can improve communication with machines using natural languages, particularly in low-resource languages.

1 Introduction

The web hosts a vast amount of information, including news articles, blogs, social media platforms, Wikipedia, and other knowledge bases. The diverse population using this e-content, encompassing different languages and age groups, creates a demand for applications with native language interfaces to access these information sources. Question Answering (QA) systems play a significant role in addressing this demand by retrieving answers for natural language queries. India, as the second most populous nation, has officially recognized 121 distinct modern Indian languages (Joshi, 2011), and users of natural language interfaces prefer accessing applications in their native languages. A survey found that Indian language internet users face challenges due to limited digital content and support in their languages (KPMG, 2017). There is a dearth of application, and services in languages that have minimal digital presence and lack annotated corpora. Moreover, Indian languages are morphologically rich, exhibit flexibility in word order and possess a complex system of post-positions. There have been fewer efforts dedicated to the QA task in several Indic languages. For open-domain QA, the task of answer retrieval holds significant importance. This article introduces a novel *kāraka*-based answer retrieval approach for QA in Indian languages, demonstrating that *kāraka* annotation captures text semantics at a level that can facilitate tasks such as answering questions and performing simple inferences. To understand the role of *kāraka*s in the answer retrieval task, let's examine the following hypothetical question-answering scenario, with a question and two possible candidate sentences for choosing the answer:

Question: Who created first effective covid-19 vaccine in India?
Possible Answer 1: First effective covid-19 vaccine in India was created by Bharat Biotech.
Possible Answer 2: The government in India created awareness regarding the administration of the first effective covid-19 vaccine.

To address the aforementioned question, methods relying solely on word overlap or answer type would be inadequate to distinguish the answer sentence effectively. In this situation, obtaining a meaning representation from the surface form text, including the event and the different participants involved, along with their respective roles, would be beneficial. A system that can assign meaningful representations to diverse inputs that share similar or common contextual knowledge, independent of specific words or sentence structures is crucial. This is shown in the example below where the action and its direct participants involved in accomplishing it are labeled with their corresponding semantic roles in both the question and the candidate answer sentences. The first sentence in the example exhibits a higher similarity in terms of the argument's semantic roles from the question, making it an appropriate answer.

Proceedings of Recent Advances in Natural Language Processing, pages 1216–1224
Varna, Sep 4–6, 2023
https://doi.org/10.26615/978-954-452-092-2_129

Question: Who$_{[AGENT]}$ created$_{[ACTION]}$ first effective covid-19 vaccine$_{[GOAL]}$ in India?
Possible Answer 1: First effective covid-19 vaccine$_{[GOAL]}$ in India was created$_{[ACTION]}$ by Bharat Biotech$_{[AGENT]}$
Possible Answer 2: The government$_{[AGENT]}$ in India created$_{[ACTION]}$ awareness$_{[GOAL]}$ regarding the administration of the first effective covid-19 vaccine.

Like this, for a QA system, understanding natural language utterances from the limited surface form involve dealing with a wide range of complex subject matters. Literature highlights language-specific resources like PropBanks (Palmer et al., 2005), FrameNets (Baker, 2014), and NomBank (Meyers et al., 2004) developed for the task of identifying how different participants associate with events. These resources have been extended to a few languages mostly because each of these framework requires its process for corpus generation that is distributed across various resources. Obtaining a comprehensive meaning representation for low-resource languages presents significant challenges due to extensive data requirements for training and evaluation.

This research addresses the challenge by employing a set of fundamental and deep semantic roles known as "*kāraka*s" that were first identified by ancient Indian grammarian Panini for Sanskrit during 4$^{\text{th}}$ century BC that symbolize the most widespread and concise form of speech during his era. By identifying the direct participants engaged in the action, *kāraka*s effectively captures the fundamental meaning of utterances. These can be applied across various languages, even those with distinct grammatical structures, resulting in an abstraction that aligns with the cognitive processes of ordinary speakers, emulates their inference methods, and enables seamless interactions with machines through query-based interactions. Additionally, it is observed that *kāraka*s can be extracted from the surface form text based on syntactic and morphological information, without the need of any extra-linguistic real-world knowledge; thus resulting in a scheme immensely valuable for low-resource languages.

The remaining article is structured as follows: Section 2 presents a concise overview of Indic QA development. Additionally, it presents a summary of NLP applications that demonstrate the usefulness of *kāraka* relations. Section 3 provides details on the proposed *kāraka*-based answer retrieval. Section 4 outlines the experiment designed to validate the proposed approach, details on the dataset used, the evaluation metrics, and the result analysis. In section 5, the paper concludes and summarizes the main findings of the research.

2 Literature Survey

The origins of the Indian QA system can be traced back to the early 2000s when Hindi-English cross-lingual QA became feasible (Sekine and Grishman, 2003). Another system used relational databases and keywords to convert user queries into SQL queries and present answers in the user's native language (Reddy et al., 2006). Several other approaches were proposed, including a natural language interface to relational databases using Paninian grammar and *kāraka*s (Gupta et al., 2012), the use of Universal Networking Language (UNL) for representing the meaning of text in the source language without translation (Shukla et al., 2004), and rule-based systems for Hindi QA (Sahu et al., 2012). Additionally, there were developments in web-based QA systems (Stalin et al., 2012), pattern matching algorithms for QA (Gupta and Gupta, 2014), question classification models (Banerjee and Bandyopadhyay, 2012), answer sentence selection models for QA (Verma et al., 2021; Joshi et al., 2022) and deep learning-based frameworks for cross-lingual (Gupta et al., 2018) and multi-lingual QA (Gupta et al., 2019). Recent experiments explored the use of transformer models pre-trained on multiple languages, with a focus on Hindi and Tamil QA, achieving improved performance in extractive QA tasks (Thirumala and Ferracane, 2022; Namasivayam and Rajan, 2023). A summary of question answering task for Indic languages is presented in Table 1. Despite efforts to develop Indic QA systems, progress may have been slower when compared to English or other widely spoken languages. With the growing emphasis on regional languages and the rapid advancements in NLP and AI, investigation on efficient QA using smaller lexicons and language models that could have broad application potential is just in time. Next section presents a summary of NLP applications demonstrating utility of *kāraka*s.

Panini identifies six *kāraka*s in Aṣṭādhyāyī, the Sanskirt monograph to express the relationship between various syntactic constituents in a sentences. *Kāraka*s account for the grammatical categories of

Reference	QA Task	Dataset Source	Size of Dataset	Domain	Approach
Kumar et al. (2005)	Hindi Closed-Domain QA	Hindi Unicode documents on agriculture and science from LTRC	30 questions	Agriculture , Science	Rule-Based: Keyword based question classification and similarity heuristics for answer extraction
Reddy et al. (2006)	Telugu Closed-Domain QA	Railway Domain	95 questions	Railway	Rule Based: Keyword and Template Based answer generation
Banerjee and Bandyopadhyay (2012)	Bengali Question Classification	Web and human annotator	1100 questions	Education, Geography History, Science from BCSTAT.COM	Data-driven: Naive Bayes, Decision Tree
Sahu et al. (2012)	Closed-Domain Hindi QA	Web and human annotator	60 questions	Not Specified	Rule Based: Lexical Similarity based answer extraction
Stalin et al. (2012)	Hindi Extractive-QA	Not Specified	5 stories, 20 questions each	Not Specified	Rule Based: Lexical Similarity based answer extraction
Gupta and Gupta (2014)	Punjabi Closed-Domain QA	Web	40 documents	Sports	Rule Based- Pattern Matching
Dua et al. (2013)	Hindi Knowledge-Based QA	Not Specified	100 questions	Not Specified	Rule Based- Dictionary Based Lookup
Kumal et al. (2014)	Hindi Knowledge-Based QA	Not Specified	240 questions	Employee Pay-roll, Enquiry, Student database	Rule Based- Dictionary Based Lookup
Seena et al. (2016)	Malayalam Closed-Domain QA	Not Specified	Not Specified	Kerela Sports	Keyword and Rule Based
Nanda et al. (2016)	Hindi Open-Domain QA	Not Specified	75 questions	Not Specified	Data-driven : Naïve Bayes
Gupta et al. (2018)	Hindi-English Multi-lingual QA	250 English and 250 Hindi documents from web	5495 questions	Tourism, History, Diseases, Geography, Economics, Environment	Deep Neural Network: CNN-RNN Based question classification, similarity computation and scoring based answer ranking
Gupta et al. (2019)	Hindi-English Cross-lingual QA	MMQA, SQuAD	MMQA-5495 questions and Translated SQuAD-18454 questions	Tourism, History, Diseases, Geography, Economics, Environment	Deep Neural Network : Attention based RNN
Thirumala and Ferracane (2022)	Hindi, Tamil Extractive-QA	Kaggle competition-chaii: Hindi and Tamil QA-Wikipedia	740-Hindi questions , 364- Tamil questions	Common	Data-driven: Pre-trained transformer models
Namasivayam and Rajan (2023)	Hindi, Tamil Extractive-QA	Wikipedia	chaii-740 Hindi questions, MLQA-5000 Hindi question, chaii-364 Tamil questions	Common	Data-driven: Pre-trained transformer models

Table 1: Summary of Indic Language Question Answering Task

the words that occur within the sentences and the role of these words within the given context, acting as a via media between the lexical/grammatical expression on one side and their semantics. Table 2 lists the six main *kāraka*s, their labels as per the popular Paninian grammar-based treebank and semantic description. Several other followers and interpreters of the Paninian grammar while studying the linguistic phenomenon in Sanskrit highlight the significance of *kāraka*s in yielding the verbal interpretation of a sentence (Kak, 1987; Bhatta, 1991; Joshi, 1991; Houben, 1997; Jyothitmayi, 2011; Kulkarni, 2021). *Desika*, the earliest prototype system developed for Sanskrit by Ramanujan (1992) elucidated that Pāṇinī's Aṣṭādhyāyī represents a grammar with extremely concise and logically coherent rules for generating accurate words and sentences in Sanskrit. This aspect might be of interest to various fields, such as computer science and artificial intelligence, due to its logical design, formalism, and well-structured arrangement of rules. Bharati et al. (1994) developed a *kāraka* parser for machine translation from Hindi to Telugu and language assessor systems (Bharati et al., 2003) from Telugu, Kannada, Marathi, Bengali & Punjabi to Hindi. Similarly, other researchers presented various machine translation systems employing *kāraka*s (Manning and Rao, 2010; H S and Idicula, 2017; Goyal and Sinha, 2009). *Kāraka*s were also utilized in word sense disambiguation (Singh and Siddiqui, 2015), text summarization for Malayalam (Kishore et al., 2016), and processing natural language queries for database extraction (Gupta et al., 2012; Gorthi et al., 2014; Jindal et al., 2014; Kataria and Nath, 2015). Additionally, researchers proposed natural language generation (Madhavan and Reghuraj, 2012), semantic role labeling (Anwar and Sharma, 2016), language encoders for vision-and-language tasks (Gorthi and Mamidi, 2022) and argument classification in Hindi-English code-mixed tweets (Pal and Sharma, 2019), all utilizing *kāraka*s. *Kāraka*s demonstrated promising results as features for argument classification, showing a strong correlation with PropBank semantic roles (Vaidya et al., 2011). *Kāraka*s have also been studied in the context of automatic question generation (Anuranjana et al., 2019). We earlier attested the utility of *kāraka*s as similarity measures in Hindi and English extractive QA systems (Verma et al., 2021), and compared them to other known similarity features. Therein we generated a feature representation for the entire passage by employing various similarity measures. The highest accuracy for selecting the best answer sentence in Hindi was achieved when combining the *kāraka* features with cosine similarity and context word overlap. Cosine similarity was computed based on vectors derived from large pre-trained models, that have limited availability. In this research, we investigate an alternative method that relies solely on *kāraka*s for initial answer sentence classification and then employs the likelihood score for ranking and answer sentence selection.

3 *Kāraka*-based Answer Sentence Retrieval

The task of answer retrieval holds significant importance in a QA system. When presented with a natural language query and a collection of sentences derived from an information extraction system, the answer retrieval module is responsible for identifying the appropriate phrase/sentence that precisely provides the answer to the user's query. The problem at hand involves a question q and a document or context (passage) containing multiple candidate answer sentences (s_1, s_2, ..., s_n) for the question. Our primary goal is to locate the most appropriate sentence, denoted by s_i (where $1 \leq i \leq n$). If the identified sentence corresponds with the actual answer, then we deem the question q to be correctly answered. We do not consider the real world scenario of unrestricted questions. To accomplish this task, we employ a supervised learning approach that treats the task as a classification problem. The diagram presented in Figure 1 illustrates the modules utilized in the *kāraka*-based answer retrieval process.

3.1 Pre-processing

Each instance in an extractive QA dataset consists of (question, context, answer) instances. We separate the context into sentences and convert the dataset into (question, sentence, target) instances. The target is a boolean value that indicates whether the sentence is an answer to the given question.

3.2 Feature Representation

For training the answer sentence classifier model, every (question, sentence) pair within the pre-processed dataset is represented using *kāraka*-based feature vector. For obtaining a *kāraka*-based feature map, every question and candidate sentence is annotated with the action verb and *kāraka*s. Additionally based on the question word, the occurrence of a specific post-position in the candidate sentence is checked using a set of hand-crafted rules. The sentence containing matching action verbs and *kāraka* arguments with the question along with the expected post-position will possess greater semantic relevance in answering the question. Thus, a (question, sentence) pair within the dataset is represented using a feature vector, corresponding to similar action verb and *kāraka*s, as well as post-position value. For identification of *kāraka* arguments to measure similarity between question and a candidate sentence, following two approaches are compared (only in resulting answer selection accuracy):

1. Data-driven *kāraka* annotator utilizing a *kāraka* annotated dataset.

2. Universal Dependency(UD) parser and UD to *kāraka* mappings.

3.2.1 Data-driven *Kāraka* Extractor

*Kāraka*s typically occur between the nominal argument and predicate within a sentence. Panini's Sanskrit grammar specifies rules to map post position of nominal and verb to *kāraka* relations between them. However, when one tries to use a rule-based system like (Bharati and Sangal, 1993; Sangal and Chaitanya, 1995; Katyayan and Joshi, 2021) for mapping from grammatical categories to *kāraka* relations, for any modern Indian languages in the same family, one faces many challenges. In this work, we implement a *kāraka* extractor based on a *kāraka* classifier model trained in a supervised manner using a Paninian dependency treebank for Hindi that includes sentences annotated with *kāraka* relations. The development process of the *kāraka* annotator is described below:

1. Every sentence in the treebank is shallow parsed to extract all noun and verb chunks.

2. The head word from the noun chunk is paired with the head word of the verb chunk, provided noun chunk occurs to the left of the verb chunk

3. Features like post-position, person, gender, number, embedding of the nominal arguments and tense, aspect and mood of the verb are extracted.

4. Every categorical feature extracted in the above step is encoded into a numeric value.

5. A training set comprising of feature representation of the identified noun-verb pairs and their target value corresponding to the *kāraka* label fetched from the treebank is prepared. If *kāraka* relation does not exist between the pair, the target value is marked as 'NA'.

6. A *kāraka* classifier is trained in a supervised manner using an artificial neural network.

Label	Name	Semantic Description	Analogous Thematic Roles
k1	*karta*	locus or source of the activity implied by the main verb	agent, causer
k2	*karma*	destination or goal of the result implied by the action	patient, goal
k3	*karna*	the means or instrument utilized for accomplishing the action	instrument
k4	*sampradana*	recipient or experiencer of the result of the object of action	beneficiary, recipient
k5	*apadana*	source of separation or point of departure	source
k7	*adhikarana*	locus of the *karta* or *karma* in time or space	location

Table 2: Six *Kārakas* from Paninian Grammar

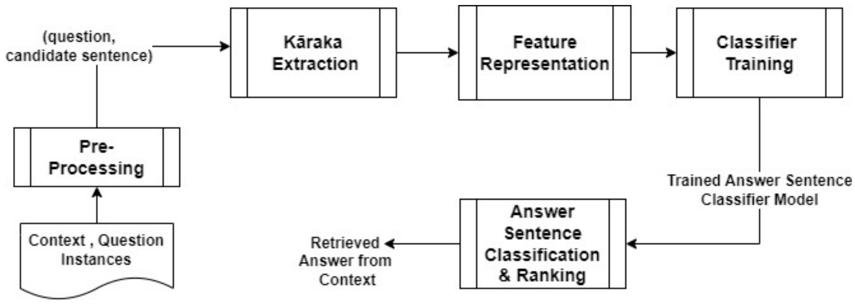

Figure 1: High Level Schematic of *Kāraka*-Based Answer Retrieval

7. The trained *kāraka* classifier model from above steps is used to predict a *kāraka* label between a candidate noun-verb pair and thus utilized for annotating a sentence with *kāraka* relations

3.2.2 Universal Dependencies(UD) to *Kārakas*

Another *kāraka* extraction technique through a mapping from UD to *kārakas* was proposed earlier (Verma et al., 2021). Therein, the UD to *kāraka* mapping was based on the study conducted for Hindi by Tandon et al. (2016). We evaluate and assess this method for *kāraka* extraction on answer retrieval accuracy on a larger benchmark corpus, as we describe the results in the latter sections below.

3.3 Classifier Training and Answer Retrieval

Using the *kāraka*-based features set a binary answer sentence classifier model is trained in a supervised manner. For differential analysis, we train two answer sentence classifier models using two different training sets, each prepared using the above two *kāraka* extraction approaches.

For answer sentence retrieval, each sentence in a context is fed into a trained model that predicts a score, indicating the likelihood of it being the answer to the question. Further, all sentences within a context are ranked in the decreasing order of the prediction scores. Based on the rank of the actual answer sentence, system performance is evaluated.

4 Experiment Design & Result Analysis

4.1 Dataset

Multilingual question answering (MLQA) (Lewis et al., 2019) is a benchmark extractive QA dataset consisting of (contexts, question, answer) pairs. Based on the number of sentences in the given context, we utilize around four thousand Hindi MLQA instances from corpus for experimental evaluation. Further, we also translated four thousand English instances from MLQA to Marathi using a model trained for English to Marathi translation on a large parallel corpora by Ramesh et al. (2022). The translation model has achieved competitive performance on the majority of datasets and has surpassed all open source publicly available models as well as commercial systems. We follow a 80:20 train:test set split for validation and evaluation for the proposed *kāraka* based approach for answer retrieval.

4.2 Implementation Details

For *kāraka* annotation using the first approach we utilized the pre-release version of Hindi treebank (Bhatt et al., 2009) for supervised learning of the *kāraka* classifier model (as discussed in section 3.2.1).

In the second approach for *kāraka* annotation through UD (discussed in 3.2.2), we employ a dependency parser developed by Qi et al. (2020). This `stanza` library offers a neural pipeline for UD parsing. To identify *kāraka* arguments from the question and answer, we utilize the UD to *kāraka* mappings presented in Tandon et al. (2016).

We train two answer sentence classifier models for Hindi using two different training sets, each prepared using the above two *kāraka* extraction approaches. For Marathi a single answer sentence classifier model is trained using the same UD to *kāraka* mappings.

Every instance in MLQA is represented using the *kāraka* based features and the answer classifier is trained in a supervised manner using a multi-layer perceptron network. The network comprises of an input layer with eight neurons, two hidden layers and an output layer with two neurons. Rectified linear activation function is used in the hidden layers. The neural network is trained using Adam optimiser and binary crossenthropy loss function.

4.3 Results and Analysis

4.3.1 Answer Sentence Classification Accuracy

For Hindi, the 10-fold cross-validation accuracy reported for answer sentence classification using UD to *kāraka* mappings is 80.17% while using our implemented *kāraka* annotator achieves 82.7%. These results highlight that for the binary answer sentence classification task both the approaches for *kāraka* extraction compare favorably for Hindi. For Marathi answer sentence classification, a comparatively less accuracy of 68.72% is reported.

4.3.2 Mean Reciprocal Rank

For further analysis, we use the Mean Reciprocal Rank (MRR) metric to evaluate the system's performance for retrieving correct answer sentences from a given context. For this we utilize the unseen test instances from dataset. This is the answer sentence selection accuracy for a given (question, context) pair. For computing this, every sentence from the given context is represented using *kāraka*-based features prepased using UD to *kāraka* approach. Further, the trained answer sentence classifier model predicts the sentence's probability score for being an answer to the given question. All sentences within a context are ranked in the decreasing order of the prediction scores. For a single query, the reciprocal rank is $\frac{1}{rank}$ where rank is the position of the actual answer sentence. For multiple queries Q, the MRR is the mean of the Q reciprocal ranks. For Hindi a MRR of 0.71 is reported while for Marathi MRR is 0.64. These results are summarized in table 3

	A	B
Hindi	80.17%	71.02%
Marathi	68.72%	64.93%

Table 3: A: Answer Sentence Classification Accuracy and B: MRR for Answer Sentence Selection using UD to *Kāraka* Mapping Approach

We observe less performance for Marathi compared to Hindi encouraging further exploration in this direction. This can be because the mapping from UD to *kāraka* for Marathi was not much refined as for Hindi. The mappings should have been obtained by generalization from several parallel (mutual translation) instances from Hindi and Marathi. Obtaining such a mapping requires careful examination of sufficiently general instances from a particular language. The drop in accuracy shows that when one tries to transfer such a mapping obtained for one language to other languages without a high level of linguistics expertise, even within the same family, one faces the challenge of choosing examples and handling exceptions. Further, the errors in the universal dependency parser can influence the overall performance of the answer retrieval. Also, the dataset used for analysis was translated from English and not a parallel corpora. For extraction of *kāraka*s, a standalone *kāraka* extractor developed in a data-driven pipeline is a better alternative to the identification of an indirect mapping through universal dependency relations that either require extensive linguistic analysis or parallel *kāraka* and UD annotated corpora for identification of mapping statistically. On the contrary, the *kāraka*-annotated corpus required for a data-driven approach to *kāraka* extraction can be developed by encoding speech patterns of a native language speaker without necessitating expert-level linguistic knowledge.

5 Conclusion

In this work, a *kāraka* based answer sentence retrieval approach is presented and its effectiveness is demonstrated through experimental analysis on a large benchmark corpus. The results clearly show that the accuracy of *kāraka* extraction impact the performance of answer retrieval, which emphasizes the need for further research and investment to enhance it. For languages lacking extensive lexical resources like PropBank and FrameNet and considering the scarcity of NLP resources for these languages, the proposed approach holds promise. We do not evaluate or dispute the usefulness of these alternative approaches. However, we observe that the identification of verb-specific semantic roles, which requires the development of comprehensive language-specific verb frames, poses a challenge, especially in low-resource languages. Our proposal is that *kāraka* annotation results into capturing semantic, facilitating tasks such as QA using smaller language models and lexicons.

References

Kaveri Anuranjana, Vijjini Anvesh Rao, and Radhika Mamidi. 2019. Hindi question generation using dependency structures. *CoRR*, abs/1906.08570.

Maaz Anwar and Dipti Sharma. 2016. Towards building semantic role labeler for Indian languages. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 4588–4595, Portorož, Slovenia. European Language Resources Association (ELRA).

Collin F. Baker. 2014. FrameNet: A knowledge base for natural language processing. In *Proceedings of Frame Semantics in NLP: A Workshop in Honor of Chuck Fillmore (1929-2014)*, pages 1–5, Baltimore, MD, USA. Association for Computational Linguistics.

Somnath Banerjee and Sivaji Bandyopadhyay. 2012. Bengali question classification: Towards developing qa system. In *WSSANLP@COLING*.

Akshar Bharati, Vineet Chaitanya, Amba P. Kulkarni, Rajeev Sangal, and G Umamaheshwara Rao. 2003. Anusaaraka: Overcoming the language barrier in india. *arXiv*.

Akshar Bharati, Vineet Chaitanya, and Rajeev Sangal. 1994. Paninian framework and its application to anusaraka. *Sadhana*, 19 Part 1:113–27.

Akshar Bharati and Rajeev Sangal. 1993. Parsing free word order languages in the paninian framework. In *31st Annual Meeting of the Association for Computational Linguistics*, volume P93–1015 of *ACL '93*, page 105–111, USA. Association for Computational Linguistics.

Rajesh Bhatt, Bhuvana Narasimhan, Martha Palmer, Owen Rambow, Dipti Sharma, and Fei Xia. 2009. A multi-representational and multi-layered treebank for Hindi/Urdu. In *Proceedings of the Third Linguistic Annotation Workshop (LAW III)*, pages 186–189, Suntec, Singapore. Association for Computational Linguistics.

V. P. Bhatta. 1991. *Epistemology, Logic, and Grammer in the Analysis of Sentence-Meaning*. Delhi, India: Eastern Book Linkers.

Mohit Dua, Sandeep Kumar, and Zorawar Singh Virk. 2013. Hindi language graphical user interface to database management system. *12th International Conference on Machine Learning and Applications*, pages 555–559.

Sai Kiran Gorthi and Radhika Mamidi. 2022. On the importance of karaka framework in multi-modal grounding.

Sai Kiran Gorthi, Ashish Palakurthi, Radhika Mamidi, and Dipti Misra Sharma. 2014. Identification of karaka relations in an English sentence. In *Proceedings of the 11th International Conference on Natural Language Processing*, pages 146–149, Goa, India. NLP Association of India.

Pawan Goyal and R. Mahesh K. Sinha. 2009. A study towards design of an english to sanskrit machine translation system. In *Sanskrit Computational Linguistics*, pages 287–305, Berlin, Heidelberg. Springer Berlin Heidelberg.

Abhijeet Gupta, Arjun Akula, Deepak Malladi, Puneeth Kukkadapu, Vinay Ainavolu, and Rajeev Sangal. 2012. A novel approach towards building a portable nlidb system using the computational paninian grammar framework. In *2012 International Conference on Asian Language Processing*, pages 93–96.

Deepak Gupta, Asif Ekbal, and Pushpak Bhattacharyya. 2019. A deep neural network framework for english hindi question answering. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.*, 19(2).

Deepak Gupta, Surabhi Kumari, Asif Ekbal, and Pushpak Bhattacharyya. 2018. MMQA: A multi-domain multi-lingual question-answering framework for English and Hindi. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Poonam Gupta and Vishal Gupta. 2014. Hybrid approach for punjabi question answering system. In *Advances in Signal Processing and Intelligent Recognition Systems*, pages 133–149, Cham. Springer International Publishing.

Sreedeepa H S and Sumam Idicula. 2017. Interlingua based sanskrit-english machine translation. In *Proceedings of International Conference on Circuit,Power and Computing Technologies (ICCPCT)*, pages 1–5.

Jan E.M. Houben. 1997. The sanskrit tradition. *The Emergence of Semantics in Four Linguistic Tradition*, pages 49–145.

Shivani Jindal, Mohit Dua, and Zorawar Singh Virk. 2014. Khik: Karaka based hindi language interface to knowledge base. In *Italian Conference on Theoretical Computer Science*.

S.D. Joshi. 1991. *Paninian Studies: Professor S. D. Joshi Felicitation Volume*. University of Michigan Press.

Shubhamkar Joshi, Jaynesh Kasliwal, Devika Verma, Namrata Kharate, and Pranali Chavhan. 2022. Empirical analysis of sentence embedding techniques for answer retrieval in marathi question answering. In *2022 Sixth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, pages 1067–1071.

Vivek Joshi. 2011. Census of India – LANGUAGE ATLAS OF INDIA 2011. Technical report, Office of the Registrar General & Census Commissioner, Ministry of Home Affairs, Government of India, India.

P.C. Jyothitmayi. 2011. The kāraka theory world of actions through words of actions. *Carmelight Journal*.

Subhash C. Kak. 1987. The paninian approach to natural language processing. *International Journal of Approximate Reasoning*, 1(1):117–130.

Aanchal Kataria and Rajender Nath. 2015. Natural language interface for databases in hindi based on karaka theory. *International Journal of Computer Applications*, 122:39–43.

Pragya Katyayan and Nisheeth Joshi. 2021. Development of automatic rule-based semantic tagger and karaka analyzer for hindi. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.*, 21(2).

Kavya Kishore, Greeshma N. Gopal, and Neethu P H. 2016. Document summarization in malayalam with sentence framing. *2016 International Conference on Information Science (ICIS)*, pages 194–200.

KPMG. 2017. Indian languages - defining india's internet.

Amba Kulkarni. 2021. Sanskrit parsing following indian theories of verbal cognition. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.*, 20(2).

Rajender Kumal, Mohit Dua, and Shivani Jinda. 2014. D-hird: Domain-independent hindi language interface to relational database. *International Conference on Computation of Power, Energy, Information and Communication*, pages 81–86.

P Kumar, S Kashyap, A Mittal, and S Gupta. 2005. A hindi question answering system for e-learning documents. *3rd International Conference on Intelligent Sensing and Information Processing*, pages 80–85.

Patrick Lewis, Barlas Oğuz, Ruty Rinott, Sebastian Riedel, and Holger Schwenk. 2019. Mlqa: Evaluating cross-lingual extractive question answering. *arXiv preprint arXiv:1910.07475*.

Manu Madhavan and P C Reghuraj. 2012. Applications of karaka relations in natural language generations. In *Proceedings of National Conference On Indian Language Computing*, pages 1–3.

Christopher Manning and Uma Maheshwar Rao. 2010. Il-ilmt sampark: A hybrid machine translation system. In *32nd All India Conference of Linguistics (AICL-32)*, pages 69–75, Lucknow, India.

Adam Meyers, Ruth Reeves, Catherine Macleod, Rachel Szekely, Veronika Zielinska, Brian Young, and Ralph Grishman. 2004. The NomBank project: An interim report. In *Proceedings of the Workshop Frontiers in Corpus Annotation at HLT-NAACL 2004*, pages 24–31, Boston, Massachusetts, USA. Association for Computational Linguistics.

Ram Vignesh Namasivayam and Manjusha Rajan. 2023. Answer prediction for questions from tamil and hindi passages. *Procedia Computer Science*, 218:1985–1993. International Conference on Machine Learning and Data Engineering.

Garima Nanda, Mohit Dua, and Krishma Singla. 2016. A hindi question answering system using machine learning approach. *International Conference on Computational Techniques in Information and Communication Technologies*.

Riya Pal and Dipti Sharma. 2019. Towards automated semantic role labelling of Hindi-English code-mixed tweets. In *Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019)*, pages 291–296, Hong Kong, China. Association for Computational Linguistics.

Martha Palmer, Daniel Gildea, and Paul Kingsbury. 2005. The Proposition Bank: An annotated corpus of semantic roles. *Computational Linguistics*, 31(1):71–106.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. Stanza: A python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 101–108, Online. Association for Computational Linguistics.

P. Ramanujan. 1992. Computer processing of sanskrit. *Computer Processing Of Asian Languages CALP-2*.

Gowtham Ramesh, Sumanth Doddapaneni, Aravinth Bheemaraj, Mayank Jobanputra, Raghavan AK, Ajitesh Sharma, Sujit Sahoo, Harshita Diddee, Mahalakshmi J, Divyanshu Kakwani, Navneet Kumar, Aswin Pradeep, Srihari Nagaraj, Kumar Deepak, Vivek Raghavan, Anoop Kunchukuttan, Pratyush Kumar, and Mitesh Shantadevi Khapra. 2022. Samanantar: The Largest Publicly Available Parallel Corpora Collection for 11 Indic Languages. *Transactions of the Association for Computational Linguistics*, 10:145–162.

Rami Reddy, Nandi Reddy, and Sivaji Bandyopadhyay. 2006. Dialogue based question answering system in Telugu. In *Proceedings of the Workshop on Multilingual Question Answering - MLQA '06*.

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

R. Sangal and V. Chaitanya. 1995. *Natural Language Procssing: A Paninian Perspective*. Prentice Hall of India.

I T Seena, G M Sini, and R Binu. 2016. Malayalam question answering system. *Procedia Technology*, 24:1388–1392.

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

Pushpraj Shukla, Amitabha Mukherjee, and Achla Raina. 2004. Towards a language independent encoding of documents: A novel approach to multilingual question answering. *In Proceedings of the 1st International Workshop on Natural Language Understanding and Cognitive Science, NLUCS*, pages 116–125.

Satyendr Singh and Tanveer J. Siddiqui. 2015. Role of karaka relations in hindi word sense disambiguation. *Journal of Information Technology Research*, pages 21–42.

Shalini Stalin, Rajeev Pandey, and Raju Barskar. 2012. Web based application for hindi question answering system. *International Journal of Electronics and Computer Science Engineering*, 2:72–78.

Juhi Tandon, Himani Chaudhry, Riyaz Ahmad Bhat, and Dipti Sharma. 2016. Conversion from paninian karakas to Universal Dependencies for Hindi dependency treebank. In *Proceedings of the 10th Linguistic Annotation Workshop held in conjunction with ACL 2016 (LAW-X 2016)*, pages 141–150, Berlin, Germany. Association for Computational Linguistics.

Adhitya Thirumala and Elisa Ferracane. 2022. Extractive question answering on queries in hindi and tamil. *ArXiv*, abs/2210.06356.

Ashwini Vaidya, Jinho Choi, Martha Palmer, and Bhuvana Narasimhan. 2011. Analysis of the Hindi Proposition Bank using dependency structure. In *Proceedings of the 5th Linguistic Annotation Workshop*, pages 21–29, Portland, Oregon, USA. Association for Computational Linguistics.

Devika Verma, Ramprasad Joshi, Shubhamkar Joshi, and Onkar Susladkar. 2021. Study of similarity measures as features in classification for answer sentence selection task in Hindi question answering: Language-specific v/s other measures. In *Proceedings of the 35th Pacific Asia Conference on Language, Information and Computation*, pages 747–756, Shanghai, China. Association for Computational Lingustics.