HKG: A Novel Approach for Low Resource Indic Languages to Automatic Knowledge Graph Construction

PREETI VATS, Department of Information Technology, Indira Gandhi Technical University For Women, Delhi
NONITA SHARMA, Department of Information Technology, Indira Gandhi Technical University For Women, Delhi
DEEPAK KUMAR SHARMA, Department of Information Technology, Indira Gandhi Technical University For Women, Delhi

Knowledge graph (KG), a visual representation of text data as a semantic network, holds enormous promise for the development of more intelligent robots. It leads to significant potential solutions for many tasks like question answering, recommendation, and information retrieval. However, this area is confined to using English text only. Since low-resource languages are now being used in the world of AI, it is necessary to develop a semantic network for them as well. In this research work, the authors provide state-of-the-art techniques for automatic knowledge graph construction for the Hindi language, which is still unexplored in ontology. Constructing a knowledge graph faces several hurdles and obstacles in the linguistic domain, primarily when it deals with the Hindi language. With an emphasis on the Indian perspective, this research intends to introduce a novel approach 'HKG' for knowledge graph construction framework for Hindi. It also implements the LSTM model to evaluate the accuracy of newly constructed knowledge graphs and compute different evaluation metrics such as accuracy and F1-score. This knowledge graph evaluates the accuracy of 87.50 using Doc2Vec word embedding with a train-test split of 7:3.

CCS Concepts: • **Computing methodologies** → **Artificial intelligence**; Natural language processing; Language resources.

Additional Key Words and Phrases: Knowledge graph Construction, Natural Language Processing, Low Resource Indian Languages, Stanza, Link Analysis and Neighbor Nodes, LSTM

1 INTRODUCTION

Knowledge graphs are innovative methods of knowledge representation. The concept of the knowledge graph was officially proposed by Google in 2012 to facilitate smart search engines [9]. After being formally introduced, the knowledge graph quickly gained popularity and raised a great deal of research discussion in academia as well as in industries. Many researchers believe that the ability of knowledge graphs to provide semantically organized information holds significant promise for developing more intelligent robots. They are now widely used in intelligent search, personalized recommendation, smart question-answering systems, anti-fraud cyber security, and other fields. Knowledge graphs also support several "Big Data" techniques in a variety of commercial and intellectual domains. While considerable progress has been made in recent years in leveraging the knowledge

Authors' addresses: Preeti Vats, preeti017phdit22@igdtuw.ac.in, Department of Information Technology, Indira Gandhi Technical University For Women, Delhi, 110006; Nonita Sharma, nonitasharma@igdtuw.ac.in, Department of Information Technology, Indira Gandhi Technical University For Women, Delhi, 110006; Deepak Kumar Sharma, dk.sharma1982@yahoo.com, Department of Information Technology, Indira Gandhi Technical University For Women, Delhi, 110006.

2375-4699/2023/8-ART $15.00
https://doi.org/10.1145/3611306

graph's outstanding capacity to deliver semantically organized information into a particular domain, there are still many unexplored possibilities.

The knowledge graph is an abstract concept that represents the real world's objects and their interconnections. Furthermore, it is a linked knowledge base made up of relationships, attributes, and entities. In a similar context, the semantic web is an earlier version of the knowledge graph [22]. The conventional approach to information retrieval has evolved due to the knowledge graph. On the one hand, Knowledge graphs define the semantic and attribute relationship between different concepts to supplement the ideas utilizing fuzzy string matching. On the other hand, knowledge graphs use a grid-based information display interface to show organized knowledge of classification and clustering. In short, knowledge graphs provide a realistic solution to the issue of manually removing irrelevant information, which is important for an intelligent society [9].

1.1 Definition of Knowledge Graph

The knowledge graph is a directed graph structure with a knowledge base that demonstrates entities and their relationships in symbolic form using the semantic network. Entities are abstract concepts or objects that occur in the real world; relationships represent the link between these entities and their attributes as their types and properties with well-defined meanings [22].

Entity1-Relation-Entity2 triplets are required to visualize the knowledge graph, where nodes stand for concepts or entities and edges for relations between them [18]. A knowledge graph is a graphical representation of context, objects, and relationships in the virtual world. There isn't a widely agreed formal definition for a knowledge graph [18].

After examining the literature by various researchers, Ehrlinger and Wöß [22] proposed Definition 1, which emphasizes knowledge graph reasoning engines. A definition of a multi-relational graph is also proposed by Wang [17] mentioned in definition 2. According to previous research, authors define a knowledge graph (KG) as

$$KG = \{E, R, F\} \tag{1}$$

where E, R, and F are sets of entities, relations, and facts, respectively. (Moreover, there is no particular well-defined definition for a knowledge graph in the digital world)

Definition 1: A knowledge graph extracts and integrates information into an ontology and applies a reasoner to derive new knowledge [8].

Definition 2: A knowledge graph is a relational graph made up of entities and relations, which are referred to as nodes and different types of edges, respectively [22].

1.2 Knowledge Graphs Applications

Researchers from the Universities of Groningen and Twente in the Netherlands coined the term "knowledge graph" for the first time in the 1980s to describe their knowledge-based system that integrates knowledge from diverse sources for modeling natural language [7, 21]. As shown in Fig. 1, where knowledge graphs are applicable in the actual world, knowledge graphs are substantial prospective solutions for various tasks like question answering, recommendation, and information retrieval, cyber security, medical, and finance among others.

A group of technologies has been referred to as knowledge graphs since 2012 as well. YAGO (Yet Another Great Ontology), DBPedia, Freebase, Google's Knowledge Vault, Wikidata, Yahoo's semantic search assistance tool Spark, Facebook's entity graph, and Microsoft's Satori are among the implementations that are frequently referenced [7, 18, 20]. It is challenging to keep an agreement on different knowledge graphs because the characteristics of these applications vary, including their architecture, operational purpose, and technological choices.

Moreover, the internet is a massive library of knowledge, where information is available in many forms (e.g., videos, photos, organized tables, etc.). However, most of the data is scattered and unstructured, making it extremely difficult to convert into a structured and machine-readable format using existing techniques[20].

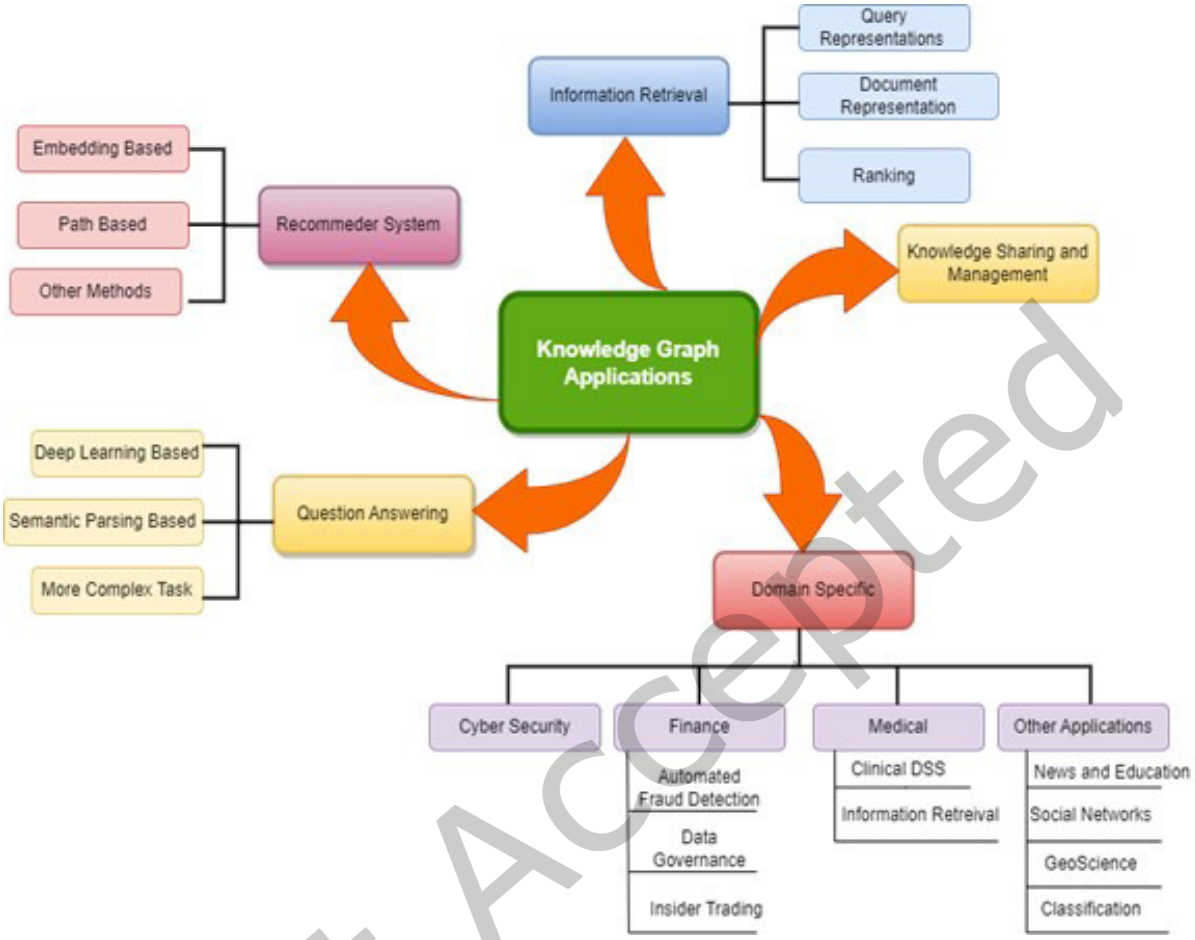

Fig. 1. Knowledge Graph Application.

Knowledge graphs aim to create extensive, organized information bases that are machine understandable. They offer structured data and factual knowledge that are utilized in several products, making them more "intelligent." Regarding search engines like Google and Bing, such knowledge graphs are used to increase the quality and relevancy of the results. Knowledge graphs are the engine of advanced AI, which enables the transformation of a single query into a continuous discussion. It helps users to converse with the system specifications and maintain context during a conversation. In the current scenario, a language-trained AI model called ChatGPT assists users by generating human-like text. For example, a user might ask ChatGPT, for instance, to "show me all the nations in the world where it's over 70 degrees Fahrenheit right now," and then, after the system responds with the information, "show me those within a two-hour flight." [17]. A user can have a complete conversational experience by going further with this.

Applications like Google Now, Cortana from Microsoft, and Siri from Apple are just a few examples of software that employs knowledge graphs to provide users with recommendations, answers to questions, and other

functions. The construction of knowledge graphs is thus a significant step toward the creation of intelligent, personalized machines. Table 1 shows a description of different knowledge graphs used by tech giants.

Table 1. Knowledge Base Models of Different Organizations

Organization	Knowledge Base Model	Data Size	Products and Development
Microsoft	Entities, their relations, and attributes	2 billion entities, 55 billion facts	ChatGPT, Lexi AI
Google	Real-world objects, relationships in a domain, and range inference	1 billion entities, 70 billion facts	Actively used in different products
Facebook	Attributes and relations are structured and strongly typed entities and indexed for efficient retrieval, search, and traversal.	50 million primary entities, 500 million assertions	Meta
IBM	Entities, and relations with context and information associated with them.	Entities >100 million, relationships >5 billion, scales on documents >100 million	IBM Watson

1.3 Motivation

Today, in the age of digital technology, the immense amount of data and information available on the internet makes most of the world's knowledge accessible. This accessibility is, however, mostly restricted to linguistic resources and extensive digital representation. Unfortunately, many languages, typically referred to as low-resource languages, have substantial disadvantages regarding online visibility and knowledge accessibility. Inadequate linguistic resources, such as extensive text corpora, lexicon databases, and other processing tools, are lacking for these languages spoken by a sizable portion of the world's population. Low-resource languages can be easily accessible using knowledge graphs.

It can be easier to document these languages by gathering and organizing language-specific data, such as lexical, semantic, and contextual knowledge. Furthermore, the availability of a knowledge graph can help to develop language technologies for low-resource languages, including speech recognition, machine translation, and natural language processing. This will increase the language's vitality in a world that is becoming more connected.

Concerning closing the digital divide and strengthening linguistic communities, constructing a knowledge graph for low-resource languages can fill the gap between digital resources and linguistic communities. This research may help to connect low-resource languages to the digital world for communication and knowledge exchange.

1.4 Knowledge Graph and Low Resource Languages

Knowledge graphs use natural language processing (NLP) and machine learning to construct a thorough representation of the nodes, edges, and labels through a process known as semantic enrichment. They can identify distinct objects and comprehend the links between them as an outcome of this process. Following this process, relevant low-resource language datasets have comparable characters compared and combined with this perspective[21].

Creating low-resource language, knowledge graphs has benefited the Semantic Web and the linguistic domain. Generally, the present search engines work on the notion of information retrieval from existing data on the Web, which includes not only files or documents but also entities, relations, and facts about things. The KG construction process employs approaches from various natural language processing (NLP) methods, machine learning, and data mining to generate a directed graph by connecting objects, relations, and facts.

As a result, the interlinked data for low-resource languages available via the Internet is sparse and insufficient. Various information retrieval methods are employed to create Indian language datasets like Hindi WordNet, Gujarati WordNet, Tamil WordNet, etc. Since most Indians speak Hindi as their mother tongue, KG for Hindi is a critical initiative, primarily for business and government purposes.

Including this, many recent studies focused on the knowledge graph (KG), which heavily relies on English consisting of different NLP techniques to acquire new knowledge. However, Hindi is more complex than other languages like English (see Section 3). Due to the inherent disparities between Hindi and English, utilizing existing NLP technologies for the English language to create Hindi KG unavoidably reaches failure. In other words, the tools employed in one language are often useless when used in another language because various NLP tools vary from one language to another due to the diverse nature of the languages. Hence, it is required to develop a framework capable of handling the Hindi language appropriately and efficiently[4].

Subsequently, the problem statement for this research is to construct a framework that can build a knowledge graph in a low-resource language, especially Hindi, without any inference of English. As all application development platforms use English translation and transliteration, it becomes imperative to develop a framework that could create a knowledge graph for low-resource languages. The objective of this research is defined as

1) Define a novel Knowledge Graph construction method for the Hindi Language.

2) Evaluate the accuracy of the proposed framework based on KG triplets.

3) Investigate the link analysis of every node to check the relationship between different entities using any language-based model.

Further, the authors succeeded in constructing a novel framework 'HKG' for Knowledge graph generation for the Hindi Language using POS tagging. To evaluate the accuracy of this framework, the authors used the Long-short term memory (LSTM) technique and tried to compile the model using DOC2Vec word embedding for triplets of KG. The results are concluded in Table 3 and 4 for further consideration. Moreover, other techniques can be used in future, to evaluate the framework, serving as the future scope of this study.

The rest of this research is organized as follows: Section 2 presents a literature review which includes the state-of-the-art of KGs to describe the foundation of KGs and related work done in this field. In Section 3, the authors presented the methodology for the construction of KGs for the Hindi language. Section 4 presents the results and discussion of the proposed framework. Challenges and future scope are presented in Section 5. Finally, the authors conclude the proposed framework and study in Section 6.

2 LITERATURE REVIEW

2.1 Foundation of Knowledge graph: State-of-Art technique

Machine learning is the classic solution to many AI problems, but learning models rely on specific training data. Some learning models can integrate with the previous knowledge base using the Bayesian structure, but these

models cannot access the information of the organized world on demand. This drawback of machine learning raised the need to extend the model including global knowledge as knowledge graph (KG) fact triplets for NLP tasks.

Knowledge graphs store information using a graph-based data model in applications that require large-scale integration, management, and value extraction from multiple data sources. Knowledge abstraction has several advantages over relational models or NoSQL alternatives using a graph-based model. Graphs provide a clear and understandable concept for a wide range of domains connected with paths represented by edges, capturing a variety of potentially complex relationships among domain components [24]. Using graphs, researchers can define a schema, allowing the data to develop more flexible databases [25]. Graph query languages provide navigational operators for locating entities connected by arbitrary-length pathways, along with the standard relational operators (joins, unions, projections, and so on) [25].

Numerous researchers and commercial enterprises have relied on the Semantic Web's characteristics in recent years to publish, parse, and analyze data effectively by machines [5]. A significant part of the information on the internet is available as plain text without any formal structure. It requires text manipulation by using machine learning tools like Natural Language Processing (NLP), Information Extraction (IE), and Information Retrieval (IR) to convert it into a structured format [5]. The main objective of the Semantic Web is to improve information integration and retrieval by transforming unstructured text into a formal representation. This concept inspired the creation of knowledge graphs (KGs), which retrieve named entities (real-world items) and their relationships from the text. Named objects and the semantic relationships between those things are typically the two primary aspects, which are used to semantically annotate and extract the text utilizing such techniques [27].

Several techniques for constructing KGs have emerged, combining discourse analysis and machine learning algorithms with pre-existing semantic frames of online data. While such methods help to process taxonomies and ontologies, they also produce many linguistic descriptions, which leads to semantic data heterogeneity and complicates data consumption [16]. The semantics of the terms used in the graph can be defined and inferred using rules and ontologies [27]. Centrality, clustering, summarization, and other functions can be assessed using scalable frameworks for graph analytics [27]. Various established and promising techniques have made it possible to use machine learning on graphs [27].

Nonetheless, there is no accepted definition of a knowledge graph (See Section 1.1) in the digital world. Instead of trying a formal definition of knowledge graphs, researchers limit themselves to a core set of traits that help us distinguish knowledge graphs from other types of information collections that would not classify as knowledge graphs [18]. Paulheim [18] defined four knowledge graph criteria. According to it, the knowledge graph is

1. discuss real-world objects and the relationships between them, visualized in a graph.
2. specifies potential entity types and relationships in a schema.
3. makes it possible to relate seemingly random entities to one another.
4. includes a range of subject areas

Furthermore, depending on the context of numerous studies, books, and other publications about knowledge graphs, an organization or community can use knowledge graphs to create a shared knowledge substrate that is constantly evolving [27]. The categorization may be either an open knowledge graph or an enterprise knowledge graph, which depends on the organization or community. Open knowledge graphs are published online, which makes it possible for users to view their content. The most famous examples, such as DBpedia, Freebase, YAGO [8], and others, cover a broad spectrum of disciplines. They offer multilingual lexicalizations (e.g., names, aliases as entities, their attributes, etc.), and are either generated by volunteer organizations [10] or directly adapted from sources like Wikipedia. There have also been publications of open knowledge graphs in various domains, including the media, politics, geography, tourism, the life sciences, and more. Typically, enterprise knowledge graphs are used internally within businesses. Enterprise knowledge graphs are frequently used for commercial

use cases and are internal to a company [5]. Web search, commerce, social networks, and finance are renowned industries that use enterprise knowledge graphs[10].

By integrating data from several sources into a new logically centralized graph-like representation, KGs achieve physical data integration as well. KGs are schema-flexible, and the graph structure makes it feasible to add new things and relate them to existing ones. Due to a lack of suitably labeled datasets, resource-constrained languages like Hindi suffer in terms of NLP applications. Though there is various triplet-based corpus available in English, they are all exclusive to the US environment and cannot work in the Indian context [19]. The aim of this study sought to propose an unsupervised method to construct a knowledge graph for a single document and address the significance of Hindi semantics to enhance the framework's quality. Conceptually, the suggested framework is built on semantic graphs [1].

In addition, SGATS (Semantic Graph-based technique for Automatic Text Summarization) is a text summarization tool for Hindi. The main objective of the research is to create a semantic graph of the original Hindi text document by using the Hindi Wordnet ontology as a background knowledge source to identify the semantic relationships between the sentences [12]. Two data sets from two separate domains, tourism, and health, were subjected to this methodology. The modern TextRank algorithm is selected to compare the performance of the suggested technique and human-annotated summary.

Furthermore, it also constitutes the text dimension which is expanded to the paragraph level by Doc2vec [6], inspired by word2vec. Doc2Vec presents two primary variants: Paragraph Vector Distributed Memory (PV-DBOW) and Paragraph Vector (PV-DM). PV-DM is comparable to Word2Vec's Skip-gram model, which predicts the current word based on the context of nearby words and a document vector. In contrast, PV-DBOW predicts words randomly from the document while ignoring the context words. Both types produce predictions by combining the document vector and word vectors. The method can then process text similarity by computing feature values like the cosine angle of the output paragraph vector. This study uses this word embedding because it works well with low-resource languages [6]. Authors used this approach for the embedding of triplets to increase the efficiency of the framework.

2.2 Related Work

An effective graph data model that supports entities and relations of various sorts and their ontological structure is required to describe and use KGs as loosely defined [11]. The graph data model should include a sophisticated query language with possibly more comprehensive graph analysis or mining capabilities, such as clustering the relevant things or identifying graph vector representation for use in machine learning tasks. Compatibility for referential integrity is also preferred because it allows for some automatic quality control of graph data by ensuring consistency. Annotating metadata of KG entities, such as information about their origin and transition during KG construction, should also be able to be represented [23].

Automatic knowledge graph construction aims to synthesize formalized human intelligence. Identification of meaningful fact patterns from different databases has traditionally entailed a substantial amount of effort. More gradually, the research [13] emphasizes acquiring conceptually structured knowledge in addition to comprehensive data. Researchers have also explored novel approaches to managing complex KGs in various contexts. Hence, a comprehensive analysis of paradigms is required to organize knowledge structures beyond data-level references [13]. Knowledge graphs are constructed in three stages: knowledge acquisition, knowledge fusion, and knowledge evolution. fig 2 also shows illustrations of all three steps, which are detailed below.

2.2.1 Knowledge Acquisition. Knowledge acquisition involves acquiring, structuring, and integrating knowledge from human experts to gather and transform problem-solving skills into a computer-readable format. The foundation of a training process is information extraction. Many NLP applications can perform knowledge acquisition tasks such as named entity recognition (NER), relation extraction, and co-reference resolution directly,

or they can supply linguistic features to other software also. Strong knowledge acquisition tool kits based on statistical methods like conditional random fields and MEM includes NLTK [11] and Stanford-NLP [15]. Moreover, these technologies can offer background functionality like NP chunks and POS tags. In other words, knowledge acquisition refers to the expertise transfer from a human expert to a machine.

2.2.2 Knowledge Fusion. The machine gathers concepts and heuristics that represent technical knowledge. According to the knowledge-fusion process, experts should "provide a framework for evaluating and incorporating new experiences and information using knowledge refinement". Knowledge refinement is the process that helps machines to build upon their fundamental knowledge using different machine-learning techniques. Knowledge refinement tools refine an existing knowledge graph by completing it or merging it with other knowledge graphs.

2.2.3 Knowledge Evolution. Recent studies have focused on how knowledge changes as a result of environmental variables. The objective is achieved by using predicated knowledge graphs to reflect facts that are tested under specific circumstances. The (h, r, t) is the definition of a conditional tuple, which can be a pre-condition triplet for a fact F (See equation 1) where h is an entity, r is a relation, and t is the time frame. Many academics have looked at this in its most basic form as a temporal knowledge graph, which is a kind of contextual information (like a timestamp) – for example, (Biden, job, vice president, 2009-2017), and (Biden, job, president, 2020-2022) illustrate a schematic of knowledge evolution.

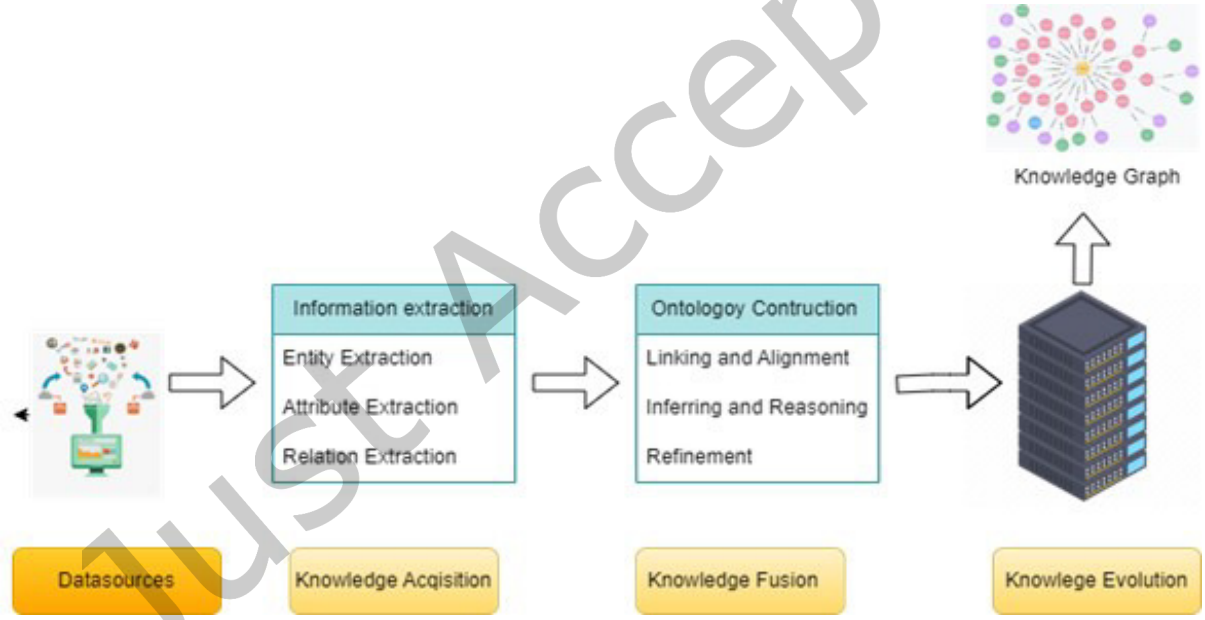

Fig. 2. Knowledge Graph Construction Phases.

Technically, strategies of knowledge graph construction are divided into four categories: (1) linguistic-based approach, (2) statistical-based approach, (3) logic-based approach, and (4) hybrid-based approach. The linguistic-based approach relies on sentence analysis, tagging of grammatical classes with their parts of speech, and syntactic-lexical patterns. The statistical-based approach works with co-occurrence analysis and knowledge-extracting methods for hierarchical linkages and rules. Inductive and logical inferences are analyzed in logic-based programming to infer or derive rules. Hybrid-based techniques typically blend conventional methods

widely used, as in recent findings such as deep learning [3]. As South Asian languages are rapidly getting into the world economy, it is mandatory to construct a graph that can represent knowledge without any translation or transliteration. In this research, the authors refer to the hybrid approach to build knowledge graphs for the Hindi story (VISHAM-SAMSYA).

Constructing lingual knowledge graphs is a long-term objective that combines unbalanced datasets dispersed across various languages. Insightful information about aligning language entities using deep learning techniques was also provided by Xlore [23]. Cross-lingual jobs continue to be severely suffered by machine translation. Initially, knowledge refinement will be hampered by faults and conflicts that occur during this translation. Second, machine learning may not have access to data resources articulated in low-resource languages. A promising direction is to resolve cross-lingual defects by accurately performing automatic low-resource knowledge graph construction.

As a result, it can be an excellent addition to already-existing, pre-trained language models. To effectively incorporate KG data into language modeling, meanwhile, is still a difficult task. Moreover, associated context is necessary for understanding any knowledge graph. A modular approach can be used to represent Hindi languages and the knowledge graph to handle KG easily [23]. While the language module provides context-aware initial embeddings for entities and relations, whereas the knowledge module creates embeddings for entities in text. These two modules work together to supply one another with crucial information. This approach can easily pre-train the model for novel knowledge graphs in low-resource languages [3].

2.3 Knowledge Graph Resources

Although all knowledge graphs depend on some given resources, based on data source type, the studies and methods of KG extraction or building may be split into 3 categories: 1) KG construction from encyclopedias and database repositories; 2) KG construction using information extraction techniques. 3) KG construction using ontologies.

2.3.1 KG creation from encyclopedias or repositories: Several repositories, like Freebase and DBpedia, have based their foundational ontologies and knowledge on the organized contents of Wikipedia pages. There have also been researched attempts made to create KG using a variety of encyclopedias, such as Wordnet [3], or encyclopedias that are not in the English language, including XLore [23], CN-DBpedi, and Zhishi.me2 [26]. Furthermore, these approaches rely on already-existing sources (such as WordNet, VerbNet, and Freebase).

2.3.2 KG construction using information Extraction Technique. Numerous studies employed Information extraction (IE) methodologies such as TEXTRUNNER. TEXTRUNNER uses deep learning methods to extract entities, properties, relations, rules, and facts from unstructured data for KG. Knowledge graphs can be constructed easily through the tagging, detection, and annotation phase of the information extraction (IE) process. After that, knowledge triplets are formed and represent the KG by linking the components and establishing facts.

2.3.3 KG construction by ontology. Numerous studies have used existing ontologies as seeds and integrated, enhanced, and filled them to create KG through platforms like NELL and Google's Knowledge Vault[2].

3 METHODOLOGIES

Overall, constructing a knowledge graph is a challenging process that requires knowledge of graph databases, natural language processing, and data management. A knowledge graph can offer helpful insights and information in various fields and applications when applied with the correct approaches and pre-processing tools. Techniques of knowledge graph construction are integrated with NLP and machine learning models. NLP procedures include pre-processing text like stop-word removal, stemming (which removes word affixes), and lemmatization (which changes words into their lemma forms). The additional NLP techniques include pattern-based,

template-based, POS tagging, and Parser. Unsupervised Machine learning models help in the training and testing of sentences for any document. This section describes the process of construction of a knowledge graph and the use of machine learning methods to evaluate their accuracy.

3.1 Data Pre-processing and Stop Word Removal

The Stanza library, formerly known as Stanford NLP [15], provides different processes for lemmatization and stemming for pre-processing Hindi text. Because none of the Python libraries support removing Hindi stop words, it must be done manually by identifying the stop words from various GitHub libraries. Hindi is a very complex language, and the resources are also limited. The mentioned approach for pre-processing helps to eliminate these types of limitations.

3.2 POS tagging and Stanza

In any language, including Hindi, sentence framing and Part-of-Speech (POS) tagging are fundamental linguistic activities. It enables people to logically portray their thoughts and emotions. The technique that assigns each word in a sentence a grammatical tag, identifying its syntactic category and role within the sentence, is known as part-of-speech (POS) tagging. The components of speech, such as nouns, verbs, adjectives, adverbs, pronouns, etc., are represented by POS tags [14]. POS tagging is an important stage in natural language processing (NLP) jobs since it aids in comprehending the grammatical structure and meaning of a sentence.

In this research, the Stanza library is employed for Hindi text having different POS tags. It supports NLP for more than 53 languages, which include languages from South Asia. It includes stemming from earlier steps, tokenization, lemmatization, and POS tagging. Moreover, it allows rendering Hindi text using several TTF files, such as Nirmala.ttf, Mangal.ttf, and Gargi.ttf. This methodology relies heavily on evaluating the Part of Speech tagging of various entities present in the story. Data preprocessing is to remove different symbols used in Hindi, such as "|," rather than glyphs (' ि ') that, when used, create several meanings for the same word. Table 2 shows the different POS tags for Hindi words with their examples.

Well-framed sentences make conversation easier to understand. This forms the basis for efficient communication. They support the clear, organized communication of concepts, ideas, and knowledge to others. Well-written sentences help users to convey the intended meaning in written texts or spoken interactions. Correct POS tagging can also aid in navigating Hindi's complicated morphology, where words can take on several forms depending on gender, number, tense, and case.

Fig 3 depicts the relationship between any two entities extracted using POS tagging. It helps to understand the grounded difference between the structure of Hindi and English sentences using the different speech formats. Sentences in English such as "I'm eating the food. " Here, 'I' and 'food' are two noun words that are mentioned at both ends of the sentences, and 'am eating' is the relation between both, mentioned between both entities. The same sentences are translated as 'मैं खाना खा रहा हूँ' in Hindi, where 'मैं' and 'खाना' are entities an 'खा रहा हूँ' is the relation mentioned at the end of the sentences. Similarly, in Fig. 4, the verbs and propositions are used to create a relation between two entities in English, but in Hindi, verb prepositions are used after mentioning both noun forms.

One-to-One word alignment is shown in Fig. 4 to explain the differences in sentence formation in both languages.

3.3 Word Embedding and Low- dimensional Vector Representation

Doc2Vec is an embedding technique that learns fixed-length vector representations for text fragments such as sentences, paragraphs, or complete documents, often known as word embeddings or document embeddings. It is a variant of the well-known Word2Vec algorithm, which comprehends word embeddings. Doc2Vec aims to

Table 2. Part-of-Speech (POS) Tagging Examples in Hindi Language

S.No.	Abbreviations	POS Tagging	Examples in Hindi
1	CC	Coordinating conjunction	लेकिन, या, फिर भी, इसलिए, जैसा
2	DD	Cardinal digit	I, II, III, IV
3	DT	Determiner	यह, ये, वह, वो
4	EX	Existential there	वहाँ है/हैं, वहाँ था/थे, वहाँ होगा।
5	FW	Foreign word	so because
6	IN	Preposition/subordinating conjunction	ने, को, के लिए, से
7	JJ	Adjective	पवित्र, स्वादिष्ट, मूल्यवान, गंभीर, महत्वकांक्षी
8	JJR	Comparative adjective	व्यापक, शांत, सस्ता, क़ीनर, करीब, बादल
9	JJS	Superlative adjective	तेज, सबसे जल्द, सबसे बड़ी, सबसे लंबा
10	LS	List Marker1	., @, ||
11	LS	List marker2	&, %, $
12	MD	Modal	चाहिए, करता था, चाहूँगा
13	NN	Singular Noun	बाग, आम, पेड़
14	NNPS	Proper noun, plural	कुर्सियों, दोस्तों, कुत्ते, घोड़ों
15	PDT	Predeterminer	सब, दोनों, आधे
16	POS	Possessive	मालिकाना, स्वत्वबोधक, अंकुश रखने वाला
17	PRP	Personal pronoun	मैं, हम, आप, वह
18	PRP	Possessive Pronoun	मेरा, हमारा, तुम्हारा, उसका
19	RB	Adverb	चलना, दौड़ना, बढ़ना

S.No.	Abbreviations	POS Tagging	Examples in Hindi
20	RBR	Comparative adverb	धीरे–धीरे, तेज
21	RBS	Superlative adverb	सबसे तेज, सबसे सुंदर
22	RP	Article	यह, मैं, है, हालांकि
23	UH	Interjection	वाह!, गजब!, बहुत बढ़िया!
24	VB	Verb	सोना, जागना, पढ़ना, घूमना
25	VBT	Verb past tense	गया, खाया
26	VBG	Verb present participle	जाना है, खाने
27	VBN	Verb past participle	मत खाओं, थका हुआ पाया
28	VBP	Past participle	पार करते हुए देखा, दौड़ रहा है
29	VBZ	Verb, 3rd person sing. present	देगा, होगा, होगी
30	WDT	Wh-determiner	कहाँ से, किसलिए, कितनी, क्या
31	QF	Quantifier	थोड़ा, कुछ
32	VM	Main verb	जाना, आना, खेलना, लेना, देना
33	PSP	Postposition, common in Indian languages	अनुगामी, अनुसरण, अनुकाल, अनुकुल
34	DEM	Demonstrative, common in Indian languages	ये, वे

learn document embeddings that capture the semantic meaning of the complete document, whereas Word2Vec focuses on learning word embeddings in the context of their neighboring words. Each document is represented by the approach as a fixed-length vector, which may then be utilized for natural language processing (NLP) tasks like document classification, clustering, or information retrieval. It is advantageous to use Doc2Vec vector representation for this research endeavor as authors generate knowledge graphs of a Hindi story that is already documented [6].

Doc2Vec can be beneficial for Hindi text for several reasons:

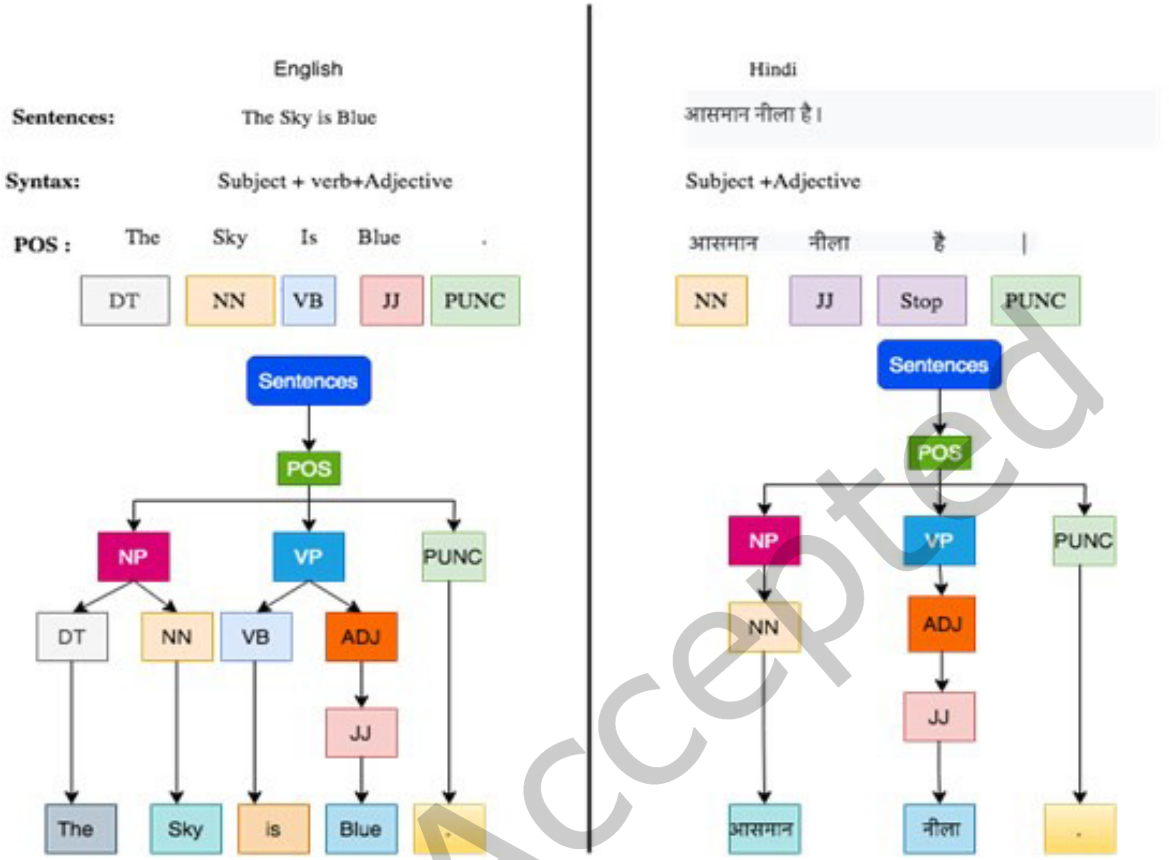

Fig. 3. POS Tagging of a sentence in English and Hindi

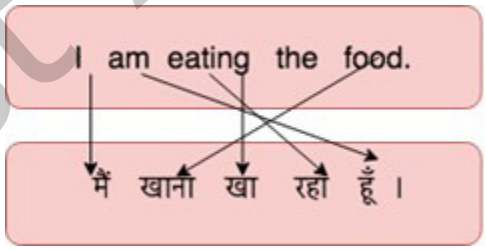

Fig. 4. One-to-one word alignment between Hindi and English

3.3.1 Semantic Presentation: Doc2Vec attempts to capture every aspect of the document's semantic meaning. Like any other language, Hindi has its meaning, nuances, and context. Doc2Vec can learn representations that reflect these nuances, enabling more accurate Hindi text analysis and modeling.

3.3.2 Contextual information and word order: Hindi is a highly inflected language with flexible word order. The ordering and context of words can significantly impact the meaning of a sentence. Doc2Vec considers the order

of words within a document and learns vector representations that incorporate this information. This allows the model to capture the relationships and dependencies between words, which is crucial for accurate understanding and analysis of Hindi text.

3.3.3 Out-of-Dictionary (OOD) words: Hindi has a vast vocabulary, and there might be instances where the model encounters words it hasn't seen during training. Doc2Vec can handle OOV words by assigning them vector representations based on their context within the document. This allows the model to generate meaningful embeddings even for previously unseen words, improving its generalization capability.

3.3.4 Document-level semantics: Doc2Vec is particularly suitable for capturing document-level semantics in Hindi text. This is beneficial when analyzing longer texts, such as articles, essays, or news reports. By learning document embeddings, Doc2Vec can represent the overall theme, sentiment, or topic of the document, enabling tasks like document classification, clustering, or information retrieval.

3.3.5 Transfer learning: Doc2Vec allows for transfer learning, where a pre-trained model on a large corpus of Hindi text can be fine-tuned or used as a feature extractor for downstream tasks. This can be helpful when working with limited labeled data for specific tasks in Hindi NLP, as the pre-trained embeddings can provide a strong foundation and improve performance.

However, it is important to maintain that the performance of Doc2Vec or any other word embedding technique is dependent on the quality and amount of the training data, as well as the specific task at hand. Doc2Vec's training strategy involves iteratively updating the word and document vectors with techniques such as stochastic gradient descent. Once trained, the resulting document vectors can be used to quantify document similarity, perform document categorization, or as input features for downstream machine learning models. To use Doc2Vec in practice, popular NLP tools such as 'gensim' in Python are used, which provides an implementation of the method. A Doc2Vec model can be trained and document embeddings for Hindi texts obtained by supplying a collection of labeled documents.

3.4 Knowledge graph construction Technique

KG construction is a linked data-mining technique that explicitly considers links between objects for building predictive or descriptive interconnected data models. Entity extraction via POS tagging, entity resolution, and relationship semantics are the three main objectives for KG creation for Hindi languages, according to [1].

3.4.1 Entity Extraction using POS Tagging: There are undersized web resources found on POS tagging with Hindi word segmentation in international research databases. Glyphs and characters in Hindi are distinct from those in other languages. It uses a unique POS tagging process, and it is exceedingly challenging to determine where POS tag limits lie [14]. In general, only 34 POS tags (refer to Table 2) are mentioned for Hindi text to train and test the proposed framework, based syntactically rather than semantically [14].

3.4.2 Entity Resolution: It refers to record links, data duplication, instance matching, object identification, and object resolution.

3.4.3 Relation Semantics: The objective of this method is to use semantic equivalents to determine whether objects in relational data are members of the same underlying entities. It attempts to extend feature-based grouping or clustering of entities with their links by taking into account both the similarity of linkages between entities and their qualities.

Including this, Graph convolutional networks are used to model the corresponding relationships between elements. GCNs achieve optimum performance using limited model complexity and only pre-aligned entities in their dataset. Based on GCN, algorithm 1, 'Build KnowledgeGraph," is designed to create a graph between

different entities based on their relations. This algorithm is generalized and can be further used to construct different language knowledge graph.

Algorithm 1: Build Knowledge Graph

Function `BuildKnowledgeGraph(`*input_data*`)`**:**

`// Step 1: Entities and attributes extraction`
Input : input_data
Output : Entities and their attributes

Relevant entities and their attributes identified using techniques such as named entity recognition (NER), part-of-speech tagging, etc.;

Extracted entities and attributes are stored in a data structure such as a dictionary or a table.;

`// Step 2: Relationships Identification and extraction`
Input : Extracted entities and attributes
Output : Extracted relationships

Techniques such as dependency parsing, co-reference resolution, and semantic role labeling are used to identify the relationships between entities.;

Extracted relationships are stored in a data structure such as a list of tuples, where each tuple contains the names of two entities and the type of relationship between them.;

`// Step 3: Knowledge graph Construction`
Input : Extracted entities and relationships
Output : Knowledge graph

A graph data structure is used (e.g., a directed graph or an RDF graph) to represent the knowledge graph.;

Each Node represents an entity and edges represent the relationships between entities.;

Metadata is associated with each node and edge to capture additional information about the entity and relationship.;

`// Step 4: Visualize knowledge graph`
Input : Knowledge graph

Visualization tools such as networks are used to create a graphical representation of the knowledge graph.;

Visualization helps to explore the structure of the graph and gain insights about the entities and their relationships.;

`// Step 5: Update the knowledge graph`
Input : Knowledge graph, New data

Periodically, the knowledge graph must be updated with new data to keep it accurate and up-to-date.;

Fig. 5 illustrates the knowledge graph formation of two different sentences in Hindi and English after data pre-processing and entity extraction

3.5 Knowledge Graphs and LSTM

Furthermore, there has been significant effort in neural machine translation, just as there has been in knowledge graphs. Here, the authors are directed to use the LSTM method to analyze the performance of the framework. LSTM (Long Short-Term Memory) architecture is a type of recurrent neural network (RNN) architecture that is

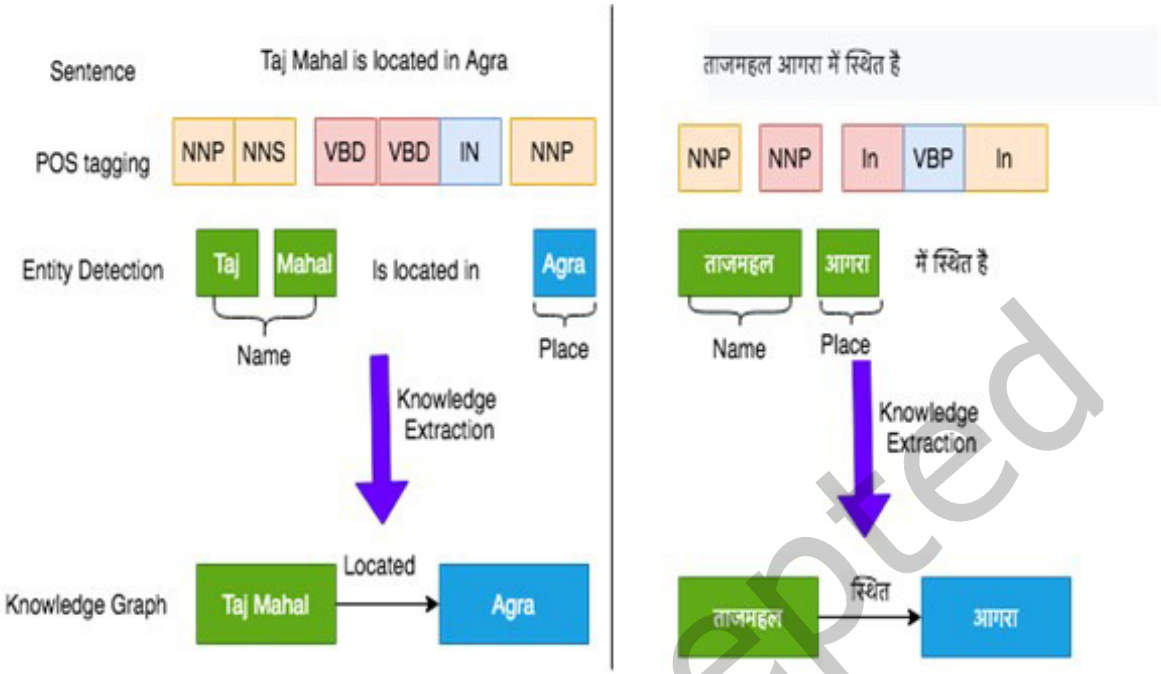

Fig. 5. Knowledge graph for English and Hindi sentences

commonly used for sequential data processing, including natural language processing (NLP) tasks like language modeling, text categorization, and machine translation. While LSTM models are most typically employed with English text, they can also be utilized with Hindi text. Depending on the preferred task, this often entails stacking one or more LSTM layers with extra layers such as thick layers or attention mechanisms. Section 4 demonstrates the results and discussion of the proposed approach.

4 RESULTS AND DISCUSSIONS

This section explains the results of the proposed approach. The authors succeeded in constructing a framework that can simplify text for a story like "Visham Samsya " by Premchand. It contains numerous entities as well as various relationships between them. Entities, their attributes, and their relations are extracted from a text file. Fig. 6 depicts a word cloud for various entities that occurred in the story. The number of entities varies roughly from 1750 to 1850. The relations are constructed between different entities based on the POS tag for the verb 'VB'.

4.1 Knowledge Graph Framework

The proposed framework 'HKG' is discussed here which can construct KG automatically for Hindi text without any inference of the English language. Fig. 7 visualizes the knowledge graph for the mentioned story, 'VISHAM-SAMASYA' as output. It is constructed using different entities, and their relationships occur in the story. 84 triplets (Entity1-relation-Entity 2) are formed using entity extraction and relationship extraction techniques. To extract triplet from text, numerous methods have recently been developed. Although these methods are effective in extracting triples from text, they still have difficulties when it comes to mapping a triple's elements,

Fig. 6. Word Cloud for entities extracted from the story: Visham-Samsya

particularly the relation, to the corresponding elements in a KG. Because it can decrease the heterogeneity of the data and improve searchability over a knowledge graph, mapping the components of a triplet to its identity is essential. Many studies have concentrated on matching up entities, which are often the subject and object of triples, with their KG counterparts.

As HKG is introduced, an automatic knowledge graph is constructed for the mentioned Hindi story. HKG is a novel approach that combines the directed graph approach with text triples, using a vector-based similarity metric and embedding vectors to identify corresponding links or relations in natural-language. Using that, the model can learn to produce similar vector embeddings for similar entities in different languages, which can be a future scope for the proposed model.

4.2 Doc2Vec: Tagged document and Word Embedding

Doc2Vec can efficiently process Hindi sentences by tranatural language-length vector representations, or embeddings. In the context of the Hindi language, these embeddings enable different natural language processing (NLP) tasks such as sentence similarity, sentiment analysis, text categorization, and information retrieval by capturing the semantic meaning of the complete sentence.

Each sentence in Doc2Vec must be assigned a distinct tag or label. During training, these tags act as identifiers for the sentences. Different tags can be applied to Hindi sentences based on sentence indices or any other suitable system. The Doc2Vec model modifies the word and phrase vectors during training to enhance the predictions. Based on the discrepancy between the expected and real words, it iteratively changes the vector representations using optimization techniques such as stochastic gradient descent.

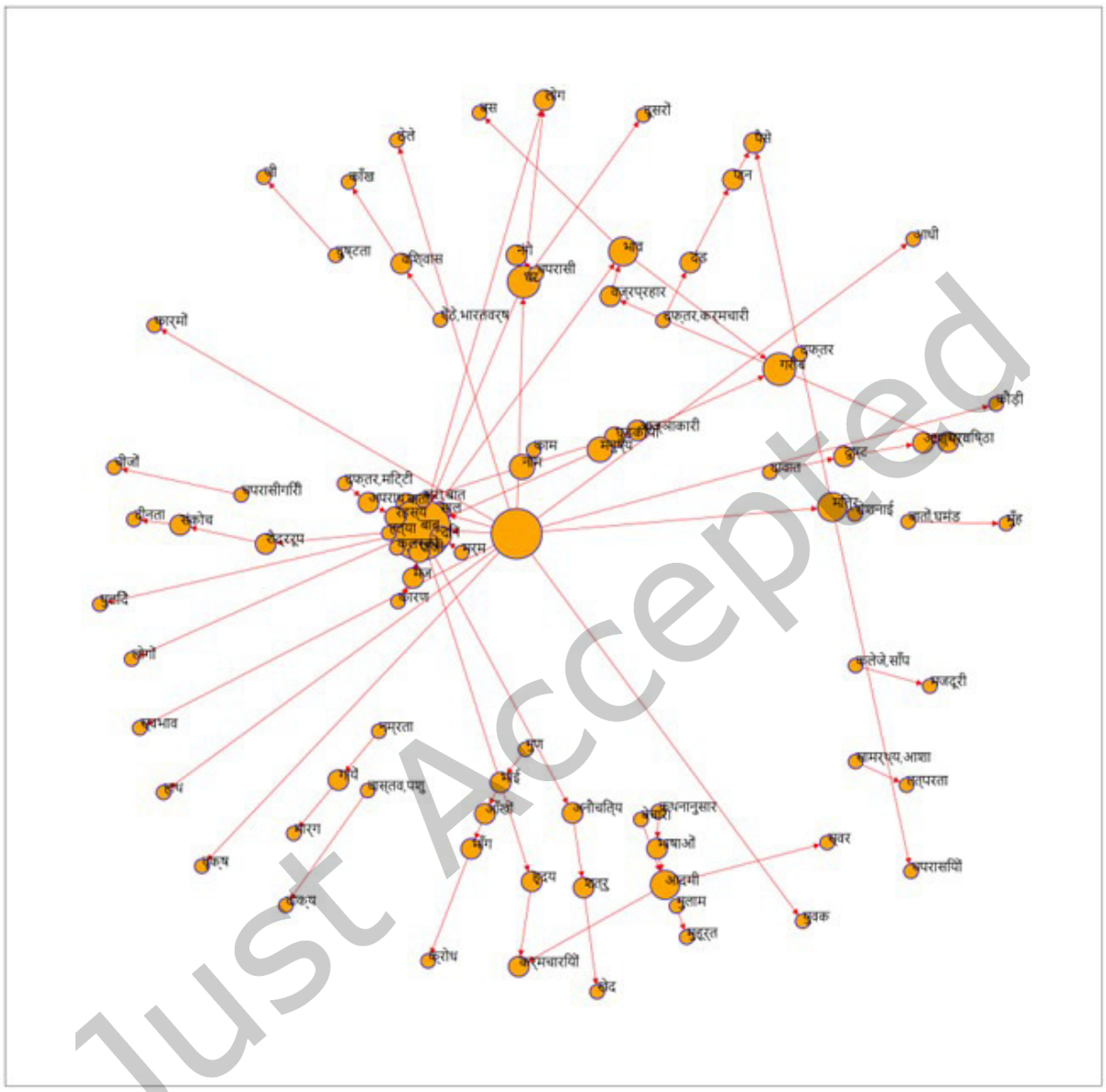

Fig. 7. Knowledge graph Visualization for the story: Visham -Samsya: Visham-Samsya

After training, the Doc2Vec model can be used to infer vector representations for Hindi sentences. A fresh sentence is run through the trained model during inference to produce a fixed-length vector that represents the semantic meaning of the phrase.

4.3 Long-Short-Term Memory (LSTM) for Hindi Text

For sequential data, such as time series or spoken language, recurrent neural networks of the LSTM (long-short-term Memory) model are frequently used. It is efficient to utilize LSTM for specific applications, such as estimating the likelihood of a link between two nodes in the graph based on their qualities, even though it is not a common method for working with knowledge graphs. The representation of each node as a series of attribute vectors, each of which represents the node's attributes at a distinct time step, is one technique to employ LSTM for a knowledge graph application.

The Knowledge Graph data would need to be preprocessed into a format appropriate for input to an LSTM network before being able to execute this strategy. This could entail extracting and transforming the properties for each node at each time step into a sequence of vectors. The LSTM architecture includes the number of layers, the number of hidden units inside each layer, and the activation functions employed, which would also need to be specified.

Using a suitable loss function and optimizer, authors train the network on the preprocessed data once it has been defined. During training, the network will develop the ability to forecast each node's label based on the order of its attribute vectors. After the network has been trained, it can be used to predict the labels of new nodes by feeding it their sequences of attribute vectors. Fig. 8 shows the flow chart of the LSTM for this particular KG.

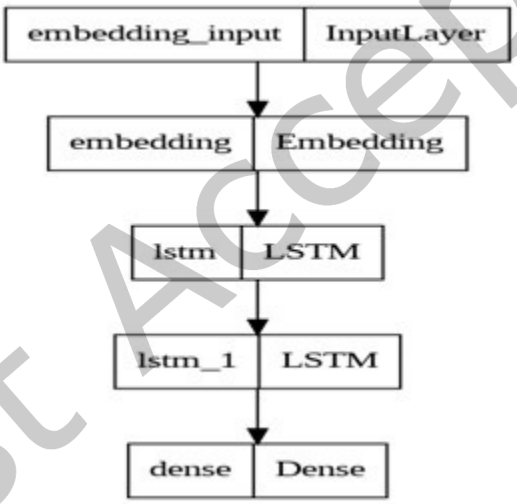

Fig. 8. LSTM Flowchart to evaluate the accuracy of the proposed framework.

Fig. 9 depicts the accuracy and loss function for the knowledge graph. In general, the accuracy of a knowledge graph depends on the quality of the data and the algorithms used to build and analyze the graph. It is important to carefully evaluate the performance of the model on a validation or test set to ensure that it generalizes well to new data. For validation, the data has been divided by 70:30 between train and test sets.

Table 3 displays the number of epochs, loss, and accuracy used by the LSTM model to evaluate the effectiveness of the knowledge graph. Here, the LSTM model is trained on the proposed knowledge graph of the described story and parameters are adjusted to assess the model's efficiency with epochs (1,50,100), respectively. The best loss function values among the three epochs were chosen i.e., Epoch 100. The effectiveness of the LSTM model is evaluated using the metric function. An optimizer function is a cost function that locates the model's best

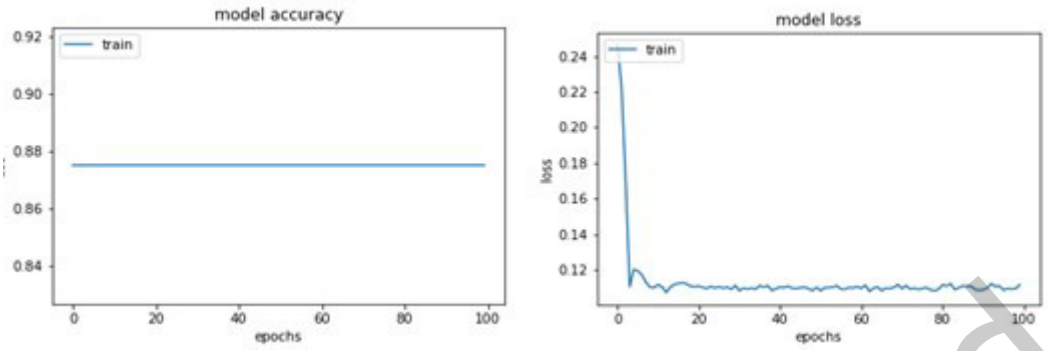

Fig. 9. Accuracy and Loss Function for LSTM network during training of proposed Framework

possible values. As it is visible that the framework has achieved the same accuracy at the mentioned epoch, but the loss decreases at every iteration. The LSTM model for the knowledge graph has been evaluated and its accuracy is found to be 87.50% during the training of the proposed framework.

Table 3. Training of Model

Epoch	Iteration	Loss	Accuracy
1	104MS/step	0.2460	0.8750
50	97MS/step	0.1103	0.8750
100	90MS/step	0.1034	0.8750

Table 4 illustrates the results of three epochs that described the testing of the LSTM model. In this case, authors tested 30% of sentences of the story, which included 45 sentence couples. Precision, recall, and F1-score were evaluated for all mentioned epochs.

Table 4. Evaluation Parameters during Epochs (Testing)

Epoch	Accuracy	Precision	Recall	F1-Score
1	0.82	0.67	0.82	0.74
50	0.82	0.68	0.81	0.73
100	0.82	0.67	0.82	0.73

5 CHALLENGES AND FUTURE WORK

KG is currently encountering numerous concerns and challenges. Some of these issues and problems are universal, while others are peculiar to Asian languages. The concerns and difficulties associated with low-resource language are discussed and described in this section [1]. Fig. 10 depicts the open and technical issues for low-resource languages discussed in subsections 5.1 and 5.2.

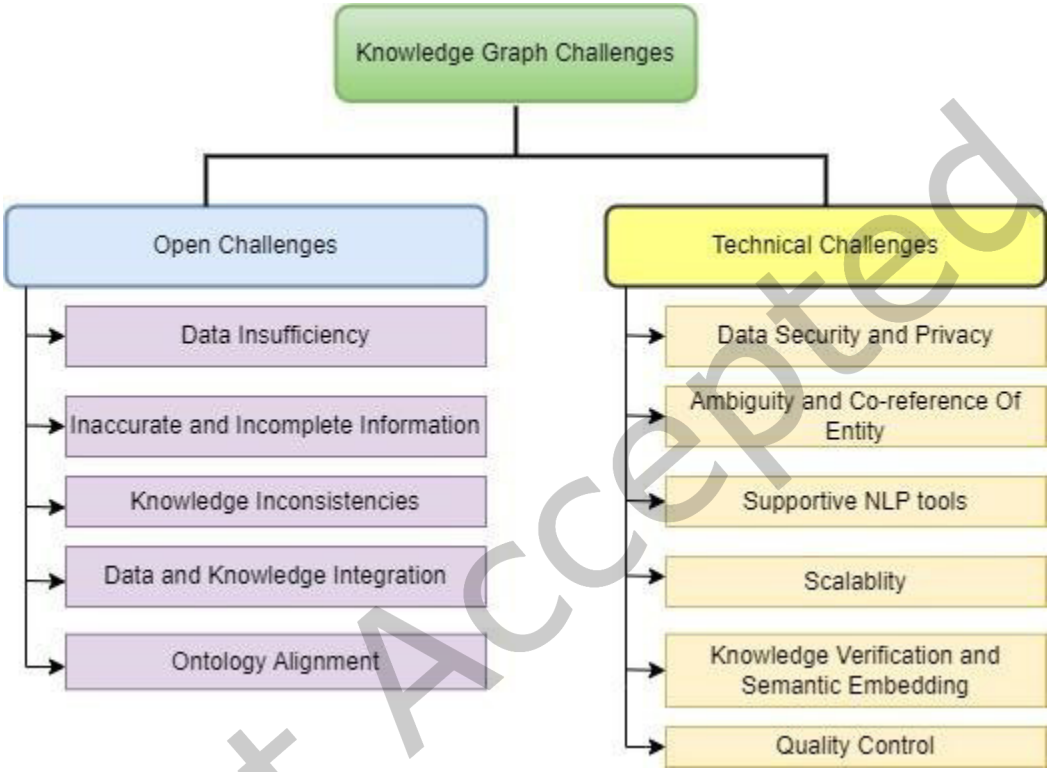

Fig. 10. Knowledge Graph challenges for Low-resource languages.

5.1 Knowledge graph open challenges for low-resource Asian Languages

A knowledge graph for the low-resource languages involves several open challenges faced by various users all over the world[11]. In this subsection, some open issues associated with Knowledge graph construction are discussed below:

5.1.1 Data Insufficiency. Inadequate data is a crucial issue in the process of getting knowledge. It is a significant open linguistic problem in terms of KG for Hindi languages when compared to KG for English languages. There are very few Asian data sources, and the vast majority of research on the issue is concentrated on geometrical areas rather than scientific topics. Moreover, there is a severe lack of training datasets [1].

5.1.2 Inaccurate and Incomplete Information. Inadequate or inaccurate information is considered as a serious problem for knowledge storage and fusion processes. It has an impact on KG's integrity and quality. Inaccurate and scattered data is one of Knowledge Graph's unresolved issues for Hindi and other low-resource languages.

While KG stores data, there are instances when some pieces are missing, and this results in knowledge being returned to users that are either partial or inaccurate. In contrast, social networks, and blogs are inundated with data. These information sources are not genuine. For knowledge acquisition and construction in the English language, numerous trustworthy sources across various disciplines can be employed in place of fake and inadequate resources. Nonetheless, the abundance of incorrect and incomplete data continues to have an impact on KG quality[11].

5.1.3 Knowledge inconsistencies. Inconsistent data is another serious obstacle to the process of knowledge fusion. The KG rules and facts for low-resource languages are often erroneous because some source data, such as social media, is imprecise or untrustworthy, or because existing extraction methods are inconsistent in extracting meaningful knowledge. This has an impact on data quality and integrity, as well as inaccurate interpretation or display of knowledge. It is worthwhile to be concerned about data inconsistency and incompleteness, which have little impact on the knowledge retrieval process.

5.1.4 Data and Knowledge Integration. Hindi language data can be gathered from a variety of sources and formats, such as organized databases, unstructured text, social media, and other verified resources. Integrating heterogeneous data necessitates dealing with a variety of data structures, schemas, and data quality challenges. During data integration, Hindi language-specific obstacles such as the use of Hindi script variations (Devanagari), transliteration issues, and regional language variances must be handled. Extraction of named entities (such as people, organizations, and places) from Hindi text is critical for knowledge graph creation, but it can be difficult due to linguistic complexity and a lack of comprehensive Hindi NLP resources.

Knowledge integrity is also recognized as a crucial issue in knowledge fusion and knowledge storage procedures. Hindi KG needs to verify that heterogeneous information has been rationally merged. Furthermore, the entities in the Hindi KG must correlate to ones in the actual world. For KG systems, knowledge integrity is a significant concern.

Methods and techniques must be developed for preserving data integrity while storing, preserving, and processing contextual information. The disparity in processing methods for Hindi and other low-resource languages, as well as the scarcity of information processing tools for Hindi, is one of KG's primary concerns for low-resource languages.

5.1.5 Ontology Alignment. Due to differences in terminology, structure, and cultural nuances, it might be challenging to align Hindi-specific ontologies with preexisting multilingual or English-centric ontologies. To provide interoperability and semantic consistency across languages, thorough mapping and alignment are required.

To appropriately describe knowledge and handle the unique qualities of the Hindi language, current ontologies may need to be enhanced or new ontologies for Hindi-specific topics may be required.

5.2 Technical Challenges for Low-resource Languages

Constructing a knowledge graph for the low-resource language also includes several technical challenges which are mentioned below:

5.2.1 Data Security and Privacy. Many Hindi data resources are used internally by organizations or authorities to protect against the exploitation or invasion of privacy of linguistic resources. But it also has an impact on the number of Language resources accessible and the learning process. Internal data retention and non-disclosure could lead to the extraction or construction of inaccurate or incomplete information. Also, one of the main issues with data repositories is the lack of secure sharing platforms for publicly disseminating Asian low-resource languages, which in turn creates a technological issue for KG.

5.2.2 Ambiguity and co-reference of Entity. Ambiguity and entity co-reference in Hindi text are similar to other languages. Ambiguity arises when a word or phrase can have several meanings, whereas coreference happens when two or more words or phrases are related to the same item. In Hindi, as in any other language, ambiguity can emerge owing to homonyms (words that sound the same but have distinct meanings), homographs (words that are spelled the same but have different meanings), and context-dependent word meanings.

Coreference in Hindi text can be established using various methods, including syntactic and semantic analysis. Pronouns, demonstratives, and other types of referring expressions can help establish coreference. For instance, consider the sentence "राम ने अपनी किताब खो दी है। वह बहुत परेशान है।" (Ram has lost his book. He is very upset.), the pronoun "वह" (he) refers back to "राम" (Ram), establishing co-reference. However, due to the language's complex syntax and morphology, coreference resolution in Hindi can be difficult. In addition, the lack of annotated corpora for coreference resolution in Hindi poses a challenge for developing effective machine learning models for this task.

5.2.3 Supportive NLP tools. The Hindi language differs from the English language in nature (see Section 3). The failure to extract and produce accurate and consistent information is caused by the absence or lack of designed instruments that are supportive of Hindi and other languages in many ways. Instead, many platforms are available to support NLP of different types of text, including Stanford NLP, the Indian Library, and iNLTK; however, occasionally it is necessary to provide a true text format (.ttf) file to support the text.

5.2.4 Scalability of Data. Large volumes of data, including text, photos, and multimedia content, must be handled by knowledge graphs to deliver full information. Scalability becomes essential for processing and storing the constantly expanding data effectively.

Fast and responsive knowledge graph operations are essential, especially when working with large datasets. This is necessary for efficient indexing, storing, and query processing strategies.

5.2.5 Knowledge verification and semantic embedding. KG systems have several barriers, but one of the most difficult is assuring the correctness, dependability, and integrity of the data. To ensure accurate and error-free knowledge storage and retrieval, flexible automatic procedures are needed. Because knowledge is now expressed using triplet form rather than multi-step relations, the issue is related to the knowledge explanation problem. Techniques that can extract entities from current knowledge and create a high-dimensional representation of them are needed to include semantic explanations for existing knowledge.

5.2.6 Quality Control. One of the most challenging issues to handle is the quality of the knowledge graph. For English, it is indeed a concern for all researchers but for low-resource languages including Hindi, it is completely neglected. Cleaning and pre-processing Hindi text data is essential to handle issues such as noise, typographical errors, inconsistencies, and linguistic variations. Spell-checking, normalization, and disambiguation techniques may be required. Resolving and linking entities across several data sources is critical for ensuring data consistency and integrity inside the knowledge graph. It is difficult to handle entity aliases, ambiguous mentions, and disambiguation.

Manual annotation and validation methods are critical for maintaining the knowledge graph's accuracy and quality. The retrieved information, entity relationships, and ontology alignments may need to be reviewed and validated by human experts in this field.

5.3 Future Scope for Low Resource Languages

The future potential of creating a knowledge graph for Hindi is tremendous, encompassing everything from linguistic resources to advanced methodologies, cross-lingual integration, and application development. Typical

knowledge graphs are static illustrations. Future research may include methods for creating dynamic and real-time Hindi knowledge graphs. These dynamic knowledge graphs may update themselves continually and adjust to shifting information sources, making them more current and pertinent. The creation of a comprehensive and well-curated Hindi knowledge graph would assist numerous sectors and contribute to the expansion of the Hindi language ecosystem. There is currently a scarcity of detailed domain-specific knowledge graphs for Hindi.

In the future, efforts could concentrate on generating graphs for specialized fields such as healthcare, economics, education, entertainment, and technology. Domain-specific knowledge graphs provide customized information retrieval, enhanced analytics, and public-specific applications in India as well as South Asia. The proposed approach of constructing knowledge graphs in Hindi creates opportunities for numerous services and applications. For instance, on top of the Hindi knowledge graph, question-and-answer systems, chatbots, recommendation engines, and intelligent virtual assistants can be developed to offer helpful information and support to Hindi-speaking users.

6 CONCLUSION

In this study, a framework that construct a knowledge graph for the Hindi story 'HKG' has been introduced, along with discussions of various KG creation methods, tasks, and phases. It also explains critical issues, obstacles, and future scope of knowledge graphs for low-resource languages. This study comprehends well for POS tagging for the Hindi language and constructs a framework that can generate knowledge graphs efficiently. This framework can be further used in different aspects of machine learning tools such as summarization, question-answering, recommender system, etc. Additionally, it also demonstrates link analysis of different triplets using the LSTM model using Doc2Vec word embedding. The accuracy of the proposed framework is evaluated and during the training and testing of the framework, encouraging results are achieved as 87.50% and 82%, respectively. This research is novel and there is neither similar work nor conflict of interest found till now.

REFERENCES

[1] Ibrahim A Ahmed, Fatima N AL-Aswadi, Khaled MG Noaman, et al. 2022. Arabic Knowledge Graph Construction: A close look in the present and into the future. *Journal of King Saud University-Computer and Information Sciences* 34, 9 (2022), 6505–6523.

[2] Zahra Zamani Alavijeh. 2015. The application of link mining in social network analysis. *Advances in Computer Science: An International Journal* 4, 3 (2015), 64–69.

[3] Saeed Albukhitan, Tarek Helmy, and Ahmed Alnazer. 2017. Arabic ontology learning using deep learning. In *Proceedings of the international conference on web intelligence*. 1138–1142.

[4] Renzo Angles, Marcelo Arenas, Pablo Barceló, Peter Boncz, George Fletcher, Claudio Gutierrez, Tobias Lindaaker, Marcus Paradies, Stefan Plantikow, Juan Sequeda, et al. 2018. G-CORE: A core for future graph query languages. In *Proceedings of the 2018 International Conference on Management of Data*. 1421–1432.

[5] Renzo Angles, Marcelo Arenas, Pablo Barceló, Aidan Hogan, Juan Reutter, and Domagoj Vrgoč. 2017. Foundations of modern query languages for graph databases. *ACM Computing Surveys (CSUR)* 50, 5 (2017), 1–40.

[6] Keliang Chen, Jianming Huang, Yansong Cui, and Weizheng Ren. 2023. Research on Chinese Audio and Text Alignment Algorithm Based on AIC-FCM and Doc2Vec. *ACM Transactions on Asian and Low-Resource Language Information Processing* 22, 3 (2023), 1–22.

[7] Gutierrez Claudio and F Sequeda Juan. 2021. Knowledge graphs. *Commun. ACM* 64, 3 (2021), 96–104.

[8] Lisa Ehrlinger and Wolfram Wöß. 2016. Towards a definition of knowledge graphs. *SEMANTiCS (Posters, Demos, SuCCESS)* 48, 1-4 (2016), 2.

[9] Xuejie Hao, Zheng Ji, Xiuhong Li, Lizeyan Yin, Lu Liu, Meiying Sun, Qiang Liu, and Rongjin Yang. 2021. Construction and application of a knowledge graph. *Remote Sensing* 13, 13 (2021), 2511.

[10] Aidan Hogan, Eva Blomqvist, Michael Cochez, Claudia d'Amato, Gerard De Melo, Claudio Gutierrez, Sabrina Kirrane, José Emilio Labra Gayo, Roberto Navigli, Sebastian Neumaier, et al. 2021. Knowledge graphs. *ACM Computing Surveys (Csur)* 54, 4 (2021), 1–37.

[11] Ali Hur, Naeem Janjua, and Mohiuddin Ahmed. 2021. A survey on state-of-the-art techniques for knowledge graphs construction and challenges ahead. In *2021 IEEE Fourth International Conference on Artificial Intelligence and Knowledge Engineering (AIKE)*. IEEE, 99–103.

[12] Manju Lata Joshi, Nisheeth Joshi, and Namita Mittal. 2021. SGATS: Semantic Graph-based Automatic Text Summarization from Hindi Text Documents. *Transactions on Asian and Low-Resource Language Information Processing* 20, 6 (2021), 1–32.

[13] Zhiwei Luo, Rong Xie, Wen Chen, and Zetao Ye. 2018. Automatic domain terminology extraction and its evaluation for domain knowledge graph construction. In *Web Intelligence*, Vol. 16. IOS Press, 173–185.
[14] Aibek Makazhanov and Zhandos Yessenbayev. 2016. Character-based feature extraction with LSTM networks for POS-tagging task. In *2016 IEEE 10th International Conference on Application of Information and Communication Technologies (AICT)*. IEEE, 1–5.
[15] Christopher D Manning, Mihai Surdeanu, John Bauer, Jenny Rose Finkel, Steven Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations*. 55–60.
[16] Jose L Martinez-Rodriguez, Ivan López-Arévalo, and Ana B Rios-Alvarado. 2018. Openie-based approach for knowledge graph construction from text. *Expert Systems with Applications* 113 (2018), 339–355.
[17] Natasha Noy, Yuqing Gao, Anshu Jain, Anant Narayanan, Alan Patterson, and Jamie Taylor. 2019. Industry-scale Knowledge Graphs: Lessons and Challenges: Five diverse technology companies show how it's done. *Queue* 17, 2 (2019), 48–75.
[18] Heiko Paulheim. 2017. Knowledge graph refinement: A survey of approaches and evaluation methods. *Semantic web* 8, 3 (2017), 489–508.
[19] Richa Sharma and Arti Arya. 2023. LFWE: L inguistic F eature Based W ord E mbedding for Hindi Fake News Detection. *ACM Transactions on Asian and Low-Resource Language Information Processing* (2023).
[20] Frans N Stokman and Pieter H de Vries. 1988. Structuring knowledge in a graph. In *Human-Computer Interaction: Psychonomic Aspects*. Springer, 186–206.
[21] Sanju Tiwari, Fatima N Al-Aswadi, and Devottam Gaurav. 2021. Recent trends in knowledge graphs: theory and practice. *Soft Computing* 25 (2021), 8337–8355.
[22] Quan Wang, Zhendong Mao, Bin Wang, and Li Guo. 2017. Knowledge graph embedding: A survey of approaches and applications. *IEEE Transactions on Knowledge and Data Engineering* 29, 12 (2017), 2724–2743.
[23] Zhigang Wang, Juanzi Li, Zhichun Wang, Shuangjie Li, Mingyang Li, Dongsheng Zhang, Yao Shi, Yongbin Liu, Peng Zhang, and Jie Tang. 2013. XLore: A Large-scale English-Chinese Bilingual Knowledge Graph.. In *ISWC (Posters & Demos)*. 121–124.
[24] Marcin Wylot, Manfred Hauswirth, Philippe Cudré-Mauroux, and Sherif Sakr. 2018. RDF data storage and query processing schemes: A survey. *ACM Computing Surveys (CSUR)* 51, 4 (2018), 1–36.
[25] Guohui Xiao, Linfang Ding, Benjamin Cogrel, and Diego Calvanese. 2019. Virtual knowledge graphs: An overview of systems and use cases. *Data Intelligence* 1, 3 (2019), 201–223.
[26] Bo Xu, Yong Xu, Jiaqing Liang, Chenhao Xie, Bin Liang, Wanyun Cui, and Yanghua Xiao. 2017. CN-DBpedia: A never-ending Chinese knowledge extraction system. In *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*. Springer, 428–438.
[27] Barry J Zimmerman. 1990. Self-regulated learning and academic achievement: An overview. *Educational psychologist* 25, 1 (1990), 3–17.