
KNOWLEDGEREDUCE: BUILDING STACKABLE SETS OF KNOWLEDGE *

Lindahl, Nels
nels.ai
Denver, Colorado
nelsl@nelslindahl.com

ABSTRACT

In an era overwhelmed by copious and complex data, the paper introduces "KnowledgeReduce," a groundbreaking conceptual framework inspired by the MapReduce model, specifically designed for constructing knowledge graphs from extensive, diverse datasets. This innovative approach reimagines MapReduce's core principles to meet the unique challenges of knowledge graph creation, such as entity recognition, relationship extraction, and data integration. KnowledgeReduce as a conceptual framework is designed to efficiently map raw data into structured entities and relationships, reducing these into a comprehensive graph format. This design framework offers a scalable, robust solution for knowledge graph construction, enhancing the processing of heterogeneous data while ensuring the reliability and relevance of the resulting graph.

Keywords KnowledgeReduce · MapReduce · Knowledge Graphs · Data Integration · Entity Recognition · Relationship Extraction

1 Introduction

In the era of prolific astroturfing, data proliferation, and content flooding the efficient organization and interpretation of vast information has become paramount, making a method to organize knowledge graphs an invaluable tool. This paper introduces KnowledgeReduce, a novel framework inspired by the MapReduce model, specifically tailored for constructing knowledge graphs from large, diverse datasets. Even massive open source data sets like "The Pile" become dated as time passes [1]. Addressing the unique challenges of this domain, such as entity recognition, relationship extraction, and data integration, KnowledgeReduce adapts MapReduce's principles to optimize for knowledge graph creation [2]. By facilitating scalable, accurate mapping of raw data into structured entities and relationships, followed by their reduction into comprehensive graph formats, KnowledgeReduce offers a robust solution for knowledge graph construction. This framework aims to bridge the gap in current methodologies, emphasizing scalability, flexibility, and efficiency in processing heterogeneous data, and sets a new standard in knowledge graph construction. A lot of the applications of knowledge graphs have helped define them operationally building definition during the refinement process [3].

The advent of the digital age has ushered in an era of unprecedented data proliferation, with vast troves of information being generated, collected, and stored every day. This burgeoning data landscape presents both opportunities and challenges, particularly in the realm of data organization, processing, and interpretation. Knowledge graphs have emerged as a critical tool in this context, offering a structured and intuitive way to represent and understand complex relationships within diverse data sets. We have hit a distinct point in the data management process where the foundations of facts are about to be defined. Knowledge foundations will form and become the underpinnings of future stores of facts. Like the growth of foundational large language models these stores of facts will only grow in size and scope.

Knowledge graphs, by their very nature, enable the integration of varied data sources into a coherent structure, facilitating more sophisticated analysis and insight generation. They are increasingly employed across a range of applications, from

**Citation:* This preprint paper was last updated on December 12, 2023. Lindahl, Nels. KnowledgeReduce: Building Stackable Sets of Knowledge. Pages.... DOI:000000/11111.

enhancing search engine capabilities to powering recommendation systems and advancing AI research. However, the construction of these graphs from large-scale, unstructured, or semi-structured data poses significant challenges. These include the efficient identification and extraction of relevant entities and relationships, handling data heterogeneity, and ensuring the scalability of the construction process. Ultimately, the best sources will define a foundation of facts.

MapReduce elegantly divides tasks into two distinct phases – ‘map’, which processes and transforms input data into intermediate key-value pairs, and ‘reduce’, which aggregates these pairs into a cohesive output. This model’s simplicity, scalability, and effectiveness in processing enormous datasets make it an exemplary foundation for adaptation in various domains. This framework needs to be updated in include a stack method to help organize and differentiate knowledge.

Inspired by the principles of MapReduce, this paper introduces KnowledgeReduce, a novel framework designed to tackle the specific challenges of knowledge graph construction. KnowledgeReduce reimagines the MapReduce paradigm, tailoring it to the intricacies of knowledge graph creation. The framework extends the map phase to include the extraction of entities and their relationships from diverse data inputs, while the reduce phase is adapted to aggregate these elements into a structured, interconnected graph.

This paper is structured to first review related work in the field, providing a background against which KnowledgeReduce is positioned. Subsequent sections detail the conceptual framework of KnowledgeReduce, its technical architecture, and the methodology employed in its implementation. The paper then presents a series of experiments conducted to evaluate the performance of KnowledgeReduce, followed by a discussion of the challenges encountered, limitations identified, and potential avenues for future research. The ultimate goal of this work is to contribute a robust and scalable tool to the data science and AI communities, facilitating the creation of knowledge graphs that can unlock deeper insights and more nuanced understandings of complex data landscapes.

2 Related Work

The development of KnowledgeReduce stands on the shoulders of significant prior work in two primary domains: large-scale data processing, particularly the MapReduce model, and knowledge graph construction methodologies. This section reviews these areas, highlighting key contributions and identifying gaps that KnowledgeReduce aims to fill. Work was done on workstation network sorting [4]. Some effort was completed related to metacomputing on the web [5].

2.1 MapReduce Model

The MapReduce model revolutionized the processing of large datasets by providing a simple, yet powerful framework that scales effectively across distributed systems. The model breaks down data processing tasks into ‘map’ and ‘reduce’ functions. The ‘map’ function processes input data into intermediate key-value pairs, while the ‘reduce’ function aggregates these pairs into a final output. This approach’s elegance lies in its ability to parallelize tasks, making it highly efficient for processing vast amounts of data.

Subsequent developments in this area, such as Apache Hadoop and Apache Spark, have built upon and extended the original MapReduce concept. Hadoop, an open-source framework, provides a distributed file system (HDFS) and an implementation of the MapReduce programming model. Spark offers in-memory data processing, which significantly speeds up tasks that require multiple data passes.

2.2 Knowledge Graph Construction

Knowledge graphs represent a method of structuring and interlinking data that enhances accessibility and analysis [6]. Early instances, such as the Semantic Web initiative led by Tim Berners-Lee, aimed to create a universal medium to share data across applications and enterprises [7]. More recently, commercial knowledge graphs like Google’s Knowledge Graph have demonstrated the immense value of this approach in enhancing search engine functionality [8].

Academic research in this area has focused on various aspects of knowledge graph construction, including entity recognition, relationship extraction, and data integration [9]. Notable methodologies involve natural language processing techniques for extracting meaningful data from text and machine learning algorithms for identifying patterns and relationships within datasets. However, a recurrent challenge in these methodologies is the efficient processing of large-scale, diverse data sources to build comprehensive and accurate knowledge graphs.

Despite significant advancements, existing approaches often struggle with scalability and flexibility, particularly in processing heterogeneous and unstructured data. Moreover, many methods require substantial domain-specific customization, limiting their general applicability.

2.3 Bridging the Gap: KnowledgeReduce

KnowledgeReduce is proposed as a solution that bridges the gap between the scalable data processing capabilities of the MapReduce model and the specific requirements of knowledge graph construction. By adapting the MapReduce framework to handle the intricacies of extracting and structuring data into knowledge graphs, KnowledgeReduce addresses the scalability and flexibility limitations of current methodologies. Furthermore, it offers a generalized approach that can be applied across various domains without extensive customization.

The subsequent sections of this paper delve into the conceptual framework of KnowledgeReduce, detailing how it adapts and extends the MapReduce model for the efficient construction of knowledge graphs.

3 Conceptual Framework of KnowledgeReduce

The conceptual framework of KnowledgeReduce is an innovative adaptation of the MapReduce model, specifically designed to address the complexities of constructing knowledge graphs from large and diverse datasets. This section outlines the foundational concepts of KnowledgeReduce, illustrating how it extends and modifies the traditional MapReduce paradigm to fit the unique requirements of knowledge graph creation.

3.1 MapReduce Revisited

To contextualize KnowledgeReduce, it is essential to revisit the core principles of MapReduce. MapReduce's two-phase approach - the 'map' phase, which processes input data into intermediate key-value pairs, and the 'reduce' phase, which aggregates these pairs into a cohesive output - provides a robust template for parallel data processing. This model excels in its simplicity and scalability, making it a staple in handling big data challenges.

3.2 Adapting MapReduce for Knowledge Graphs

KnowledgeReduce takes the skeleton of the MapReduce model and tailors it to the specific needs of knowledge graph construction. Knowledge graphs, unlike other data structures, require not just the processing of data, but also the understanding and representation of complex relationships between data entities. KnowledgeReduce, therefore, introduces adaptations in both the map and reduce phases:

3.3 The Mapping Phase

3.3.1 Entity and Relationship Extraction

In this phase, raw data is parsed to identify entities (nodes) and their relationships (edges). Unlike traditional MapReduce, which focuses on simple key-value pair generation, this step involves sophisticated parsing techniques, potentially incorporating natural language processing (NLP) and machine learning (ML) algorithms to accurately extract semantic relationships.

3.3.2 Intermediate Representation

The output of the mapping phase in KnowledgeReduce is a set of tuples or structured representations that encapsulate entities and their interrelations. These representations are prepared for aggregation in the subsequent phase.

3.4 The Reducing Phase

3.4.1 Aggregation and Conflict Resolution

Here, the intermediate data from the mapping phase is consolidated. This step involves aggregating information about the same entities from different sources, resolving conflicts (such as differing information about an entity from multiple sources), and refining the graph's structure.

3.4.2 Graph Synthesis

The final step of the reduce phase is the construction of the knowledge graph. This involves creating nodes and edges based on the aggregated data, ensuring that the graph accurately reflects the relationships and properties extracted during the mapping phase.

3.5 Scalability and Performance

KnowledgeReduce inherits the scalable architecture of MapReduce but introduces additional optimizations for knowledge graph construction. It leverages distributed computing resources to handle the intensive computation required for parsing and synthesizing large datasets. The framework is designed to be adaptable to different scales of data, from small corpora to extensive, web-scale datasets.

3.6 Flexibility and Applicability

A key aspect of KnowledgeReduce is its flexibility. The framework can be tailored to various domains and data types without significant restructuring. Whether the source data is structured, semi-structured, or unstructured, KnowledgeReduce's mapping phase can be adjusted to extract relevant information effectively. This flexibility makes it a versatile tool applicable in diverse fields, from biomedical research to financial analytics.

3.7 Summary

The KnowledgeReduce framework represents a novel approach to knowledge graph construction, addressing the challenges of scalability, accuracy, and flexibility inherent in processing large-scale data. By adapting and extending the MapReduce model, KnowledgeReduce provides a powerful tool for transforming raw data into structured, interconnected knowledge graphs. The subsequent sections will delve into the technical architecture of KnowledgeReduce, its implementation details, and the results of performance evaluations.

4 Data Ingestion and Preparation

A critical first step in the KnowledgeReduce framework is the ingestion and preparation of data. This phase sets the foundation for the effective functioning of the subsequent mapping and reducing phases. It involves sourcing data from various repositories, preprocessing it to a uniform format, and ensuring its quality and consistency. This section outlines the strategies and methodologies employed in KnowledgeReduce for data ingestion and preparation.

4.1 Data Sourcing

KnowledgeReduce is designed to work with a wide range of data sources, including structured databases, unstructured text files, web pages, and even streams of real-time data. The flexibility in data sourcing is crucial for constructing a comprehensive knowledge graph that encapsulates diverse knowledge domains.

4.2 Multi-Source Integration:

Structured Data: Databases, CSV files, and other structured formats are ingested using standard database querying languages (like SQL) or file-reading APIs. **Unstructured Data:** Text data, web content, and other unstructured data forms are sourced using web scraping tools, APIs, or direct file access. **Real-Time Data Streams:** For applications requiring up-to-date information, KnowledgeReduce can integrate data from streaming sources, using technologies like Apache Kafka or AWS Kinesis.

4.3 Data Preprocessing

Once the data is sourced, it undergoes several preprocessing steps to ensure consistency and quality. This stage is crucial as it directly impacts the accuracy and reliability of the knowledge graph.

4.4 Standardization and Cleaning:

4.4.1 Normalization

Data is normalized to ensure consistent formats across different sources. This includes standardizing date formats, text encodings, and numerical representations.

4.4.2 Cleaning

Data cleaning operations are performed to remove or correct inaccuracies, missing values, and irrelevant information. Techniques such as tokenization, stemming, and lemmatization are applied to text data to reduce complexity.

4.5 Schema Mapping and Entity Recognition

4.5.1 Schema Mapping

For structured data, schema mapping is employed to align different data models. This step ensures that similar entities from different sources are correctly identified as equivalent.

4.5.2 Entity Recognition

In the case of unstructured data, Named Entity Recognition (NER) algorithms are used to identify and tag entities within the text.

4.6 Ensuring Data Quality

Data quality is paramount in KnowledgeReduce. The framework implements several checks and balances to maintain the integrity of the data.

4.6.1 Verification and Validation

Automated scripts and algorithms check for data anomalies, inconsistencies, and outliers.

4.6.2 Quality Assurance

Periodic manual reviews are conducted, especially for critical datasets, to ensure the data adheres to the expected quality standards.

4.7 Scalable Data Handling

Given the potentially vast volume and variety of data, KnowledgeReduce employs scalable data handling techniques. Distributed storage systems like Hadoop Distributed File System (HDFS) or cloud-based solutions (e.g., Amazon S3) are utilized for storing large datasets. Data preprocessing tasks are parallelized where possible to optimize performance.

5 The Mapping Phase

The mapping phase is a pivotal component of the KnowledgeReduce framework, where raw data is transformed into an intermediate structure conducive to knowledge graph construction. This phase involves sophisticated data processing techniques to extract meaningful entities and their relationships from the ingested and preprocessed data. This section elucidates the methodologies and algorithms employed in the mapping phase of KnowledgeReduce.

5.1 Entity and Relationship Extraction

The core objective of the mapping phase is to identify and extract entities (nodes of the knowledge graph) and their relationships (edges). This task varies in complexity depending on the nature of the source data.

5.2 Structured Data

From structured datasets, entities and relationships are extracted based on predefined schemas. Relational databases, for example, provide clear entity relationships through foreign keys and table schemas.

5.3 Unstructured Data

With unstructured data sources, such as texts or web pages, more complex methods are employed. Natural Language Processing (NLP) techniques, such as Named Entity Recognition (NER) and Dependency Parsing, are utilized to identify entities and their interrelations within the text.

5.4 Intermediate Representation

Upon identifying entities and relationships, the data is converted into an intermediate format suitable for the reducing phase. This representation typically consists of tuples or structured objects capturing the essential attributes of entities and the nature of their relationships.

5.5 Key-Value Pairs

Each entity and its associated attributes are encapsulated in a key-value pair, where the key is the entity identifier, and the value is a composite of its attributes and relationships. Relationships between entities are similarly represented, with key-value pairs denoting the entities involved and the type or nature of their relationship.

5.6 Handling Complexity and Ambiguity

The mapping phase in KnowledgeReduce also addresses the complexities and ambiguities inherent in data interpretation.

5.7 Disambiguation

Techniques such as context analysis and cross-referencing with existing knowledge bases are used to disambiguate entities (e.g., distinguishing between 'Apple' the technology company and 'apple' the fruit). Machine Learning models, trained on large corpora, are employed to enhance the accuracy of entity and relationship extraction, especially from unstructured data. Scalability and Distributed Processing To manage the volume and velocity of data, KnowledgeReduce implements scalable mapping strategies. This includes:

5.7.1 Distributed Processing

Leveraging distributed computing resources to parallelize the mapping process, thereby handling large datasets efficiently.

5.7.2 Optimization Algorithms

Utilizing optimized algorithms for entity extraction and key-value pair generation to minimize computational overhead.

The mapping phase in KnowledgeReduce is a sophisticated process that transforms raw data into a structured format, ready for aggregation into a knowledge graph. By employing advanced NLP techniques, machine learning algorithms, and scalable processing strategies, this phase ensures that the extracted entities and relationships are accurate, relevant, and primed for the subsequent reducing phase. The next section will delve into the reducing phase, where these mapped entities and relationships are aggregated to construct the knowledge graph.

6 The Reducing Phase

The reducing phase is the transformative stage in the KnowledgeReduce framework, where the intermediate data generated during the mapping phase is aggregated, refined, and synthesized into a coherent knowledge graph. This section discusses the methodologies and strategies implemented in the reducing phase, focusing on how KnowledgeReduce consolidates the mapped data into a structured knowledge network.

6.1 Aggregation and Conflict Resolution

One of the primary functions of the reducing phase is to aggregate information about the same entities from different sources and resolve any conflicts that arise from this process.

6.2 Data Aggregation

The reducer aggregates mapped data based on entity identifiers. It combines attributes and relationships from various sources, creating a comprehensive view of each entity. In cases where an entity is mentioned in multiple sources, the reducer collates all relevant information, ensuring no data is lost in the aggregation process.

6.3 Conflict Resolution

When discrepancies arise (e.g., different sources provide conflicting information about an entity), KnowledgeReduce employs conflict resolution strategies. These strategies may include prioritizing more reliable sources, using consensus algorithms, or applying machine learning models trained to infer the most accurate information.

6.4 Graph Synthesis

With the aggregated data, the reducing phase proceeds to the synthesis of the knowledge graph. This involves the creation of nodes (entities) and edges (relationships) based on the refined data.

6.5 Node Creation

Each unique entity forms a node in the graph. The node's attributes are populated with the aggregated data, providing a multi-faceted representation of the entity.

6.6 Edge Creation

Relationships between entities are represented as edges. The nature, type, and strength of these relationships (as extracted and processed in the mapping phase) define the characteristics of the edges.

6.7 Ensuring Graph Integrity

The integrity of the knowledge graph is paramount. KnowledgeReduce implements several measures to ensure the accuracy and reliability of the graph.

6.8 Consistency Checks

Automated checks are performed to ensure data consistency across the graph.

6.9 Validation Rules

The graph is subjected to a set of validation rules that verify the correctness of relationships and attribute values.

7 Implementation and Experimentation

The practical implementation and experimental evaluation of the KnowledgeReduce framework are crucial for demonstrating its effectiveness and efficiency in real-world scenarios. This section details the implementation process of KnowledgeReduce and discusses the experiments conducted to validate its performance.

7.1 Experimentation and Evaluation

To evaluate the performance of KnowledgeReduce, a series of experiments were conducted using datasets from different domains, including healthcare, finance, and social media. The objectives of these experiments were to assess the scalability, accuracy, and speed of the KnowledgeReduce framework.

7.2 Experiment Design

Prompt: "Build python code for a knowledge reduce function designed to create stackable sets of knowledge by extending MapReduce broken down step by step."

```
# Step 1: Import necessary libraries
import mapreduce
from typing import List, Tuple, Any

# Step 2: Define the Mapper function
def mapper(document: str) -> List[Tuple[str, int]]:
    """
    The mapper function takes a document (or a part of it)
    and produces a list of key-value pairs.
    """
    # Assuming the document is a string of text, split it into words
    words = document.split()
    # Map each word to a key-value pair (word, 1)
    return [(word, 1) for word in words]
```

```

# Step 3: Define the Reducer function
def reducer(key: str, values: List[int]) -> Tuple[str, int]:
    """
    The reducer function takes a key and a list of values,
    and combines the values for that key.
    """
    # Sum up the values, which are the counts of the words
    return (key, sum(values))

# Step 4: Define the Knowledge Reduction function
def knowledge_reduce(documents: List[str]) -> List[Tuple[str, int]]:
    """
    The knowledge reduce function applies the MapReduce
    pattern to a list of documents to reduce them to a
    set of key-value pairs representing knowledge.
    """
    # Apply MapReduce
    map_reduce_result = mapreduce.map_reduce(documents, mapper, reducer)

    # Additional logic can be applied here to further
    process the results for knowledge reduction

    return map_reduce_result

# Note: This code is a basic outline and needs further
refinement based on specific requirements.
# The 'mapreduce' module used here is a placeholder
and would need to be replaced with an actual MapReduce
framework or library.

```

7.2.1 Datasets

Varied in size from gigabytes to terabytes to test scalability.

7.2.2 Metrics

Included the time taken for graph construction, the accuracy of entity-relationship mapping, and the overall integrity of the knowledge graph.

7.2.3 Comparison

Performance was compared with traditional knowledge graph construction methods to highlight the improvements offered by KnowledgeReduce.

8 Challenges and Limitations

While the KnowledgeReduce framework represents a significant advancement in the field of knowledge graph construction, its development and deployment have encountered several challenges and limitations. This section discusses these challenges, offering insights into the complexities of implementing such a framework and the inherent limitations encountered during its application.

8.1 Handling Diverse Data Formats

Addressing the challenge of handling a wide array of data formats, especially the integration of unstructured and semi-structured data, required the development of advanced parsing algorithms. These algorithms are crucial for effectively managing the diversity of data types, ensuring that the information from various sources is accurately captured and integrated into the knowledge graph. However, a notable limitation of this approach is the potential

loss of subtle nuances in the data during the parsing process. This loss can sometimes affect the overall richness and detail of the knowledge graph. The balancing act between efficiently processing a wide range of data formats and maintaining the integrity and depth of the information is a critical aspect of this endeavor. It highlights the ongoing need for improvement in parsing techniques to minimize data loss and maximize the fidelity of the knowledge graph representation.

8.2 Data Quality and Consistency

Ensuring data quality and consistency across various sources emerged as a significant challenge, particularly given the potential for inconsistencies to lead to inaccuracies in the knowledge graph. Addressing this required the implementation of sophisticated data cleaning and normalization techniques. However, in the context of large-scale datasets, the risk of residual inaccuracies persists. This situation underscores the critical need for continuous improvement in data management strategies. These strategies must not only tackle the vast scale and complexity of the data but also enhance the precision and reliability of the knowledge graphs. The ongoing development of more robust data validation and correction methods is essential to minimize errors and maintain the integrity of the knowledge graph, especially as the scale and diversity of the data sources grow.

9 Future Work and Enhancements

The development of KnowledgeReduce represents a significant step forward in the field of knowledge graph construction. However, as with any pioneering technology, there is room for further development and enhancement. This section outlines potential areas for future research and improvements that could augment the capabilities of the KnowledgeReduce framework.

10 Conclusion

The development and exploration of the KnowledgeReduce framework, as presented in this paper, mark a significant advancement in the domain of knowledge graph construction and processing. KnowledgeReduce successfully adapts and extends the foundational principles of the MapReduce model, specifically addressing the unique challenges associated with processing large-scale, diverse datasets into structured, meaningful knowledge graphs.

Throughout this paper, we have delineated the conceptual framework of KnowledgeReduce, highlighting its innovative approach to data ingestion, mapping, and reducing phases tailored for knowledge graph construction. The framework's implementation details and its experimental evaluation have demonstrated not only its feasibility but also its effectiveness in handling complex datasets with efficiency and accuracy.

One of the key strengths of KnowledgeReduce lies in its scalability and flexibility, making it a potent tool for a wide array of applications across various domains, from enhancing search engine capabilities to facilitating advanced research in fields like healthcare and environmental studies. The challenges and limitations encountered during the development and application of KnowledgeReduce have provided valuable insights, paving the way for future enhancements and research directions.

As we look ahead, the potential areas for improvement and expansion of KnowledgeReduce present exciting opportunities. Advancements in natural language processing, real-time data handling, user interface design, and integration capabilities are just a few avenues that hold promise for further elevating the utility and applicability of this framework.

As we conclude this introduction, KnowledgeReduce stands as a testament to the ongoing evolution in data processing and knowledge management. By bridging the gap between large-scale data processing and knowledge graph construction, it offers a robust, scalable, and efficient solution for transforming raw data into actionable knowledge, thereby contributing to more informed decision-making and innovative data analysis in our increasingly data-centric world.

Acknowledgments

This work product is a product of nels.ai research based on my interest in sharing and learning about machine learning and all things technology. I appreciate all of the kind words and suggestions Substack readers of The Lindahl Letter have provided over the last three years as well.

References

- [1] Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, et al. The pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*, 2020.
- [2] Jeffrey Dean and Sanjay Ghemawat. Mapreduce: simplified data processing on large clusters. *Communications of the ACM*, 51(1):107–113, 2008.
- [3] Lisa Ehrlinger and Wolfram Wöß. Towards a definition of knowledge graphs. *SEMANTiCS (Posters, Demos, SuCCESS)*, 48(1-4):2, 2016.
- [4] Andrea C Arpaci-Dusseau, Remzi H Arpaci-Dusseau, David E Culler, Joseph M Hellerstein, and David A Patterson. High-performance sorting on networks of workstations. In *Proceedings of the 1997 ACM SIGMOD international conference on Management of data*, pages 243–254, 1997.
- [5] Arash Baratloo, Mehmet Karaul, Zvi M Kedem, and Peter Wijckoff. Charlotte: Metacomputing on the web. *Future Generation Computer Systems*, 15(5-6):559–570, 1999.
- [6] Dieter Fensel, Umutcan Şimşek, Kevin Angele, Elwin Huaman, Elias Kärle, Oleksandra Panasiuk, Ioan Toma, Jürgen Umbrich, Alexander Wahler, Dieter Fensel, et al. Introduction: what is a knowledge graph? *Knowledge graphs: Methodology, tools and selected use cases*, pages 1–10, 2020.
- [7] Tim Berners-Lee, James Hendler, and Ora Lassila. The semantic web. *Scientific american*, 284(5):34–43, 2001.
- [8] Heiko Paulheim. Knowledge graph refinement: A survey of approaches and evaluation methods. *Semantic web*, 8(3):489–508, 2017.
- [9] Natasha Noy, Yuqing Gao, Anshu Jain, Anant Narayanan, Alan Patterson, and Jamie Taylor. Industry-scale knowledge graphs: Lessons and challenges: Five diverse technology companies show how it’s done. *Queue*, 17(2):48–75, 2019.