From Linguistic Linked Data to Big Data

Dimitar Trajanov[1], **Elena-Simona Apostol**[2], **Radovan Garabík**[3], **Katerina Gkirtzou**[4], **Dagmar Gromann**[5], **Chaya Liebeskind**[6], **Cosimo Palma**[7], **Michael Rosner**[8], **Alexia Sampri**[9], **Gilles Sérasset**[10], **Blerina Spahiu**[11], **Ciprian-Octavian Truică**[2], **Giedre Valunaite Oleskeviciene**[12]
[1]University Ss Cyril and Methodius in Skopje, North Macedonia, dimitar.trajanov@finki.ukim.mk
[2]National University of Science and Technology Politehnica Bucharest, Romania
{elena.apostol, ciprian.truica}@upb.ro
[3]Slovak Academy of Sciences, Slovakia, radovan.garabik@kassiopeia.juls.savba.sk
[4]Athena R.C.-ILSP, Greece, katerina.gkirtzou@athenarc.gr
[5]University of Vienna, Austria, dagmar.gromann@univie.ac.at
[6]Jerusalem College of Technology, Israel, liebchaya@gmail.com
[7]University of Naples "L'Orientale", Italy, cosimo.palma@phd.unipi.it
[8]University of Malta, Malta, mike.rosner@um.edu.mt
[9]University of Cambridge, UK, alexia.sampri@gmail.com
[10]Univ. Grenoble Alpes, CNRS, Grenoble INP, LIG, France, gilles.serasset@imag.fr
[11]University of Milan-Bicocca, Italy, blerina.spahiu@unimib.it
[12]Mykolas Romeris University, Lithuania, gvalunaite@mruni.eu

Abstract

With advances in the field of Linked (Open) Data (LOD), language data on the LOD cloud has grown in number, size, and variety. With an increased volume and variety of language data, optimizations of methods for distributing, storing, and querying these data become more central. To this end, this position paper investigates use cases at the intersection of LLOD and Big Data, existing approaches to utilizing Big Data techniques within the context of linked data, and discusses the challenges and benefits of this union.

Keywords: Linguistic Linked Open Data (LLOD), Big Data, Linguistic Data Science, efficient processing

1. Introduction

Linguistic Linked (Open) Data (LLOD)[1] applies LOD principles and Semantic Web technologies to language data, offering a standardized way of representing and sharing linguistic datasets, such as lexica, ontologies, corpora, treebanks or terminologies, in a machine-readable format. This allows such datasets to be linked and integrated across multiple resources, enabling new forms of linguistic analysis and discovery emerging from their interoperability. Many language data practitioners are creating, publishing or interlinking more and more data in the LLOD cloud[2] (Chiarcos et al., 2012), which has been growing steadily since its inception in January 2011. This increase in available data raises the potential of the LLOD cloud to address new use cases requiring the interoperability of resources which, since then, were only available in their individual data silos.

In this position paper, we claim that if we want to turn this potential ability of the LLOD cloud exemplified in the use cases into real use and applications, we will have to go beyond in-memory triple stores, single-server graph databases or federated queries to several public SPARQL endpoints and deal with scalability issues raised by the handling of the LLOD cloud as a whole. Big Data processing and analysis techniques have been proposed to address particularly large and heterogeneous data sets. The LLOD cloud is particularly large and heterogeneous due to many globally distributed small producers of language resources, each producing one corpus in one language (e.g. Mukhamedshin et al., 2020), one dictionary in several languages (e.g. Gracia et al., 2018), etc. These resources are of high quality, multilingual and multi-level in the sense of consisting of primary data, e.g. a corpus, and annotations, e.g. in form of meta-data describing specific aspects of the primary data. The more structured data are, the higher is the potential for interlinking and uncovering new information. However, current methods to query and reuse LLOD resources suffer from problems of scalability and processing speed. Thus, we argue that Big Data techniques might be a good solution for processing this particular type of linguistic data and to boost LLOD-based linguistic data science.

Literature on utilizing Big Data processing and analysis on structured data has focused on the re-

[1]"Open"is in brackets since proprietary data can also be published as linked data.

[2]The LLOD cloud (`https://linguistic-lod.org/llod-cloud`) is the set of all (interlinked) language resources made available on the web.

LREC-COLING 2024, pages 7489–7502
20-25 May, 2024.

lation to knowledge graphs, such as data storage (e.g. Chawla et al., 2020), distribution (e.g. (e.g. Chawla et al., 2021), and query optimization (e.g. Konstantopoulos et al., 2016). Janev et al. (2020b) provide an excellent overview of Big Data tools and applications in connection with knowledge graphs. Another rapidly evolving related field is that of Big and Open Linked Data (BOLD) (Janssen and van den Hoven, 2015), which unites the concepts of open data, linked data, and Big Data. However, to the best of our knowledge, this is the first publication to focus on the potential of processing the LLOD cloud with Big Data techniques, which we exemplify with three use cases: uncovering translation mappings across languages, accessing linguistic information, and extracting information.

2. Preliminaries

Depending on the theoretical foundation, research community, field and representation among other factors, data can be categorised differently. For instance, particularly large and heterogeneous data are described as Big Data, when paired with high-quality they might be called beautiful data. Representing beautiful language data as LOD is called LLOD. Our main focus is on the intersection of LLOD and Big Data, both of which we briefly introduce in this section. The base architecture of querying LLOD by means of Federated SPARQL as opposed to Big Data Apache Spark Clusters is depicted in Fig. 1.

2.1. Linguistic Linked Open Data

High-quality digital language data are vital to tasks in linguistics, information extraction, NLP among others. However, creating, linking, and re-using language data is time-consuming and challenging since they might be represented, annotated, and described with metadata from different perspectives, with varying degrees of coverage, and in different formats. The objective of LLOD (Chiarcos et al., 2011) is to establish interoperability between multilingual language data with different annotation layers from various, distributed, and heterogeneous sources by utilising the principles proposed for LOD (Bizer et al., 2009). Publishing language data as LLOD assigns global and unique identifiers to each element, which allows them to be addressed through standard Web protocols and to be uniformly linked and re-used. They are represented in the Resource Description Framework (RDF) (Cyganiak et al., 2014) format, which can be serialized in different formats from XML and JSON to Turtle, and queried with standardized query languages, especially SPARQL. The predominant model to represent LLOD is OntoLex (McCrae et al., 2017), which also represents an important mechanism to integrate resources and services into language technology pipelines (McCrae and Declerck, 2019). These data can serve as input to Large Language Model (LLM) fine-tuning and fact checking and LLOD formats can be used to structurally represent the output of LLMs. For instance, Oleškevičienė et al. (2021) analyze speaker attitude by means of discourse markers automatically detected with XLM-R and then represented as LLOD in the cloud. Comparing discourse markers across languages can uncover new knowledge and quality issues in one language, whereby the overall quality of the data for fine-tuning LLMS can in turn be improved.

2.2. Big Data

In today's world, we are experiencing unparalleled growth in data generation, a phenomenon referred to as Big Data. This surge is also evident in the field of linguistics, where datasets are growing rapidly and becoming more complex. The advent of Big Data brings unprecedented challenges in managing and analyzing vast, complex datasets (Naeem et al., 2022). Traditional tools falter with data that exceeds system RAM, demanding introduction of distributed computing across computer clusters. This shift requires rethinking the foundational principles of single-node systems. For example, distributing data across multiple nodes slows down data access and increases failure risks. Consequently, a programming paradigm aligned with the system's characteristics is essential for efficient, parallel code execution. The concept of Big Data is intrinsically linked to five core characteristics, collectively known as the "5Vs". These characteristics, which define the nature of Big Data, are volume, velocity, variety, veracity, and value (Abdalla, 2022). In terms of Big Data processing tools, Spark is the most popular according to a JetBrains report in 2022[3], with 31% of developers using it, followed by Hadoop MapReduce at 16% and Hive at 13%. For streaming processing tools, Spark Streaming leads the way with 20% of developers using it, followed by Flink at 8% and Storm at 6%.

3. Use cases

The union of Linguistic Linked and Big Data approaches can be beneficial for a large number of potential use cases from discovery of translation equivalents to crosslingual requirements engineering, with a particular focus on efficient and fast processing of distributed resources. In this section, we exemplify the potential of this union by

[3] https://www.jetbrains.com/lp/devecosystem-2022/

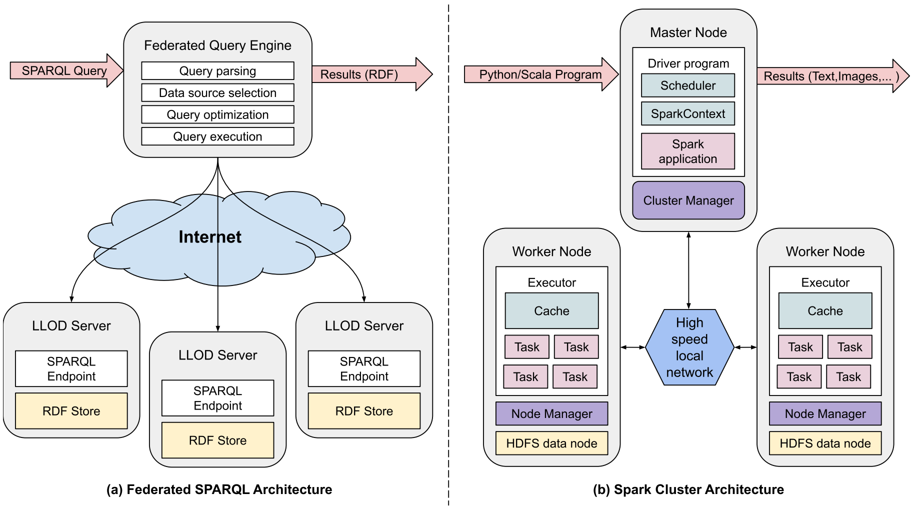

Figure 1: Architecture diagrams of (a) Federated SPARQL and (b) Apache Spark Cluster. The Federated SPARQL architecture enables querying distributed RDF data sources, while the Spark Cluster architecture is designed for processing large-scale data using the Apache Spark framework.

means of use cases where LLOD can strongly benefit from efficient Big Data processing.

3.1. Linking, Expanding and Enhancing DBnary

Wiktionary[4] is a well-known collaborative, multilingual online dictionary that provides definitions, translations, pronunciations, etymologies, and other lexical information about words in various languages. Due to its status as a vast and easily accessible lexical database, it serves as a highly valuable resource for numerous language-related tasks and applications. However, despite containing somewhat structured data, the lexical information within Wiktionary is not readily annotated in a machine-readable, formal, structured format. The desirability, and challenge, of accessing this lexical data is evident in the numerous projects aimed at parsing and extracting data from Wiktionary, which have been developed over the years of its existence.

The DBNary dataset (Sérasset, 2015) described in Sérasset (2015) is an RDF version of lexical data extracted from 23 languages editions of the Wikitionary projects. Each language edition describes lexical entries of multiple languages in the edition's language. For instance, the English language edition describes 1,217,180 English entries[5] and 6,318,874 non English entries (accounting for 3,361 languages) with definitions in English, while French edition describes 554,487 French entries and 1,090,482 non French entries (accounting for 4,678 languages) with French description.

DBnary is updated each time a new Wiktionary dump is made available by the Wikimedia foundation, hence it has a new version twice a month. From September 2012 (first extract) to April 2017, the DBnary dataset was modelled using the original lemon vocabulary (McCrae et al., 2011) and since then it uses OntoLex (McCrae et al., 2017) model. Each version is kept and made available either from Zenodo[6] (for versions up to 2017) or from the DBnary website[7] (for versions from 2017). The whole set of available dumps in BZip2 compressed format represents more than 100GB of data. The public SPARQL endpoint always reflects the latest version of the extracted data (along with a summary of all versions statistics in RDF datacube format).

Being a dataset of more than 414M triples, with a new version twice a week, DBnary by itself shows the *volume* and *velocity* core characteristics of Big Data, arguably along with *veracity* and *value*. The *velocity* of DBnary is one of its major strengths as the dataset evolves almost in real time. For instance, the term $COVID_{en}$ is available in DBnary since February 20th, 2020 while it was unavailable in almost all other datasets and was still unknown early 2023 in some of the major Large Language Models. This velocity is usually eluded, and usages we are aware of always consider one unique version as, even if it is a rather big knowledge graph, it is still manageable on a single Openlink Virtuoso[8] server node. However, this drastically

[4] https://www.wiktionary.org/

[5] All counts given in this paragraph reflect the `20230420` version of the dataset, extracted from the Wiktionary dump produced April 20th 2023.

[6] http://zenodo.org

[7] http://kaiko.getalp.org/about-dbnary

[8] https://vos.openlinksw.com/

limits the use cases of the dataset.

For instance, Chiarcos and Sérasset (2022) use DBnary to create a cross-lingual query system for DBpedia (Auer et al., 2007), by linking DBpedia concepts with DBnary terms in the user language. This work uses SPARQL federated query on DBnary and DBpedia endpoints to account for the datasets volume and only create the linking on the fly to escape from the velocity of DBnary (the queries are always performed on the latest version). If they had chosen to statically create an alignment from DBnary to DBpedia, the alignment itself would have to be performed twice a month or the DBnary version would have to be fixed a priori.

Also, Tchechmedjiev et al. (2014) showed that the DBnary dataset itself can be enhanced by disambiguating the >10M provided translation pairs, i.e., attaching the translation to a word sense rather than to a lexical entry, allowing to clearly state in which context, *bleu*$_{fr}$ can be translated to *green*$_{en}$.[9] The shortcoming of this work lies in the fact that it only disambiguates the *source* word sense of translations but does not propose a solution for the disambiguation of the *target* of the translation, hence, we cannot clearly state which word sense of *green*$_{en}$ is a valid translation of *bleu*$_{fr}$. The disambiguation of the source of translations is light enough to be performed on each version of the dataset as it can be done directly after extraction, using only data from the current language edition.[10] However, disambiguating the target of the translation is more complex and attempts that have been performed exploiting the topology of the full dataset or the computation of cross-lingual similarity measures lead to two main scalability problems. First, such methods need at least a set of fully disambiguated translations that are needed as a gold standard for intrinsic evaluation of the process. However, as the dataset is constantly evolving, with changes in definition, ordering or addition/deletion of word senses, such a gold standard has to be corrected for each extracted version, and this is already a complex task that involves dealing with two different versions and that needs to be performed twice a month. Second, in the case of cross-lingual similarity measurements, some experiments have been performed using node or sentence (definition) embeddings, but current approaches fail to scale to the size of the full dataset graph. Current experiments on such embeddings only use monolingual graphs and involve a computation time that currently forbids the disambiguation to be performed for each dataset version (twice a month).

These considerations show that if we want to extend DBnary, either with manually created data or with computed information, we need to resort to Big Data techniques both to be able to compute such data, but also to make it evolve and stay in sync with the ever-changing DBnary versions.

3.2. Accessing Corpora and Linguistic Information

With the proliferation of Large Language Models (LLMs) and Generative AIs, it is likely that the Internet will soon become inundated with automatically generated and machine translated text, hard to distinguish from human-generated content. This will significantly diminish the usefulness of new web corpora, while curated "national" corpora are likely to remain a valuable source of proven human-generated texts for the time being. However, these corpora are usually closed to outside NLP applications, and a standardized or at least a semi-standardized way of accessing the content as LOD would be a significant improvement. Ideally, the access would be in a federated manner, covering multiple sets of corpora at multiple locations, provided by separate established institutions. We are not necessarily advocating using SPARQL in lieu of the Corpus Query Language[11] (CQL), as such an implementation change would probably be a major effort.

A similar concept has been implemented in the form of CLARIN Federated Content Search[12], which defines data formats for structuring standardized query results. This system is primarily geared towards human interaction and has not gained widespread usage beyond selected corpora within the CLARIN infrastructure.

Diachronic research is seen as a specialized field, where we explicitly take into account the time dimension in the data. Big Data in the form of massive linguistic data could be used to trace semantic change, capture semantic cultural shifts, the evolution of grammar, etc. One well-known example of a diachronic resource (accessible in the form of a search engine) is the Google Ngram Viewer (Michel et al., 2011), available in several

[9]Indeed, even if *bleu*$_{fr}$ is usually translated to *blue*$_{en}$ when it denotes a color, it can also be translated to *rookie*$_{en}$ or *green*$_{en}$ when it denotes a inexperienced soldier.

[10]Translations are usually linking the language of the edition (source) to other languages (target), the process simply involves a monolingual semantic similarity measure based on a string distance method and the gold standard used to evaluate the methodology is also directly extracted from the language edition.

[11]https://www.sketchengine.eu/documentation/corpus-querying/

[12]https://www.clarin.eu/content/federated-content-search-clarin-fcs-technical-details

major languages and widely used, despite the closed nature of the data themselves. For example, Li and Siew (2022) used the English Google Ngram Corpus to extract contextual information about words for each year from 1800 to 2000. The authors used contemporary data on human processing and learning words searching for relations between semantic change and cognitive constraints. Traditionally, research on semantic change focuses on language evolution and usually searches for the patterns and laws in historical corpora (Hamilton et al., 2016). The framework of NexusLinguarum (Armaselu et al., 2022) suggests the combination of NLP and LLOD techniques for automatically detecting and representing semantic change using sources of linguistic data accessed as LLOD. Generally, diachronic research is not limited to corpora, but to any source of data with a clearly defined and accessible time dimension.

Using LLOD is not limited to a linguistic audience. As an end-user-oriented use case, we introduce the platform *Slovake.eu*, offering language courses for Slovak at different levels. The website provides a variety of exercises, tests, and dictionaries to help users familiarize themselves with Slovak grammar, learn new words, and improve their language skills. Additionally, users can interact with other learners of Slovak through the site. Apart from language courses, the portal also contains reading material (information about Slovakia, its history, geography, and some fiction) aimed to improve users' proficiency with the language. The portal is interactive, with exercises containing links to spoken sentences and a built-in multilingual dictionary. The learners can invoke the dictionary by clicking on any individual word in the teaching texts. Currently, the portal is being overhauled with the addition of new lessons covering additional proficiency levels and with a new version of the built-in dictionary. The dictionary uses DBpedia, DBnary, and Wikidata[13] to extract structural information for the word and present the relevant data (such as translation into the language of the instruction and grammatical categories) to the user in an intuitive and unobtrusive way.

The use of LLOD in this portal is a prime example of an end-user application. The portal utilizes an existing source of Big Data (i.e. DBPedia) with a clearly defined structure and access to obtain information relevant for its purposes.

3.3. Information Extraction

Although Information Extraction (IE), the task of automatically extracting structured information from unstructured documents, is by now a well-established branch of NLP, much of the work carried out has been directed towards the analysis of fixed text databases pre-established in advance of processing.

One of the defining characteristics of Big Data mentioned earlier is *velocity*. This typically applies to streamed data generated in real-time at a rate that precludes such pre-storage in one place before processing begins. A representative use-case is weather prediction which draws on information continuously arriving from thousands of weather stations, for which it has been shown that Big Data streaming techniques can be used to great advantage (see Fathi et al. (2021) for a comprehensive review). Now, such techniques have mainly been applied to numerical data.

We suggest that there exist domains for which Big Data streaming techniques could also offer advantages where the data is predominantly *linguistic*. For example, IE where predominantly textual data arrives dynamically, as when monitoring evolving news sources. A (pre-Big Data) forage into such a domain was NewsExplorer (Pouliquen et al., 2006), developed at the Joint Research Centre, Ispra, which automatically acquired knowledge by continuously analysing approximately 15,000 incoming newspaper articles per day. The system displayed evolving stories dynamically on a geographical map. Amongst the sub-services required were the identification of people, places and other named entities, computation of relationships between them, such as the most important people mentioned in the context of a certain country or issue. In addition, the source material occurred in 13+ languages, further complicating the problem of correctly linking entity mentions to their semantic referents.

More recently, Herodotou et al. (2020) real-time detection framework for aggression on Twitter data employs state-of-the-art streaming Machine Learning (ML) methods deployable on engines such as Apache Spark. Of note is the authors' claim that the framework can easily scale to increase its throughput to accommodate the entire Twitter Firehose with only a small number of commodity machines. Another use-case is the field of social influence analysis based on social networking services, such as Facebook, Twitter, and LinkedIn. All of these generate huge quantities of streamed multimodal content that includes not just text, but also images, audio, and video that is used for tasks such as extraction of popular topics, evaluation of social influence, identification of influential users, and modeling of information diffusion. Peng et al. (2017) survey mentions that the achievement of these tasks involves not only dealing with the inherent computational complexity of a social network with millions or billions of nodes but also the integration of multiple data sources with implicit con-

[13] https://www.wikidata.org

nections.

All of these examples tend to confirm that the combination of Big Data streaming with an LLOD-based representation system is a promising direction for investigating 'dynamic' IE. The key issue is how to define a set of key services (such as entity and event extraction) based on the potential for integrating different kinds of information offered by LLOD.

4. Existing LOD and Big Data Approaches

This section organizes existing LOD and Big Data approaches based on their contributions to LLOD, which include data distribution, storage, mining, integration, and query optimization.

4.1. Data Distribution

Data partitioning (or fragmentation) is employed by Big Data systems to offer improved query performance, reduce storage requirements per node, and increase scalability (Truică and Apostol, 2021). This involves splitting data into smaller shards using various configurations, including horizontal, vertical, mixed horizontal-vertical, and mixed vertical-horizontal fragmentation methods. Big Data systems use data replication to offer high availability, fault tolerance, and seamless access to data in case of downtime (Truică et al., 2015) using either primary-secondary (single primary node) or multi-primary (multiple primary nodes) configurations. In a primary-secondary configuration, the clients only interact with the primary node, synchronizing secondary nodes, while in a multi-primary configuration, clients interact with all the nodes, with synchronisation occurring synchronously or asynchronously. Synchronous replication guarantees data integrity but may impact performance, while asynchronous replication enhances performance but may risk data loss if the primary storage fails. Additionally, the Interplanetary File System (IPFS), a decentralized, peer-to-peer file system, is proposed to publish LOD (Sicilia et al., 2016), offering LOD availability, resilience, and sustainability, particularly suitable for data fragmentation and replication in BOLD systems due to its built-in decentralized distribution and deduplication capabilities.

4.2. Data Storage

Various technologies and frameworks, including Hadoop, centralized RDF stores, and in-memory stores, can be used to implement Big RDF storage solutions (Chawla et al., 2020). In the Hadoop framework, query processing options include MapReduce or Apache Spark, with data storage in Hadoop Distributed File System (HDFS) or NoSQL databases like HBase (Shvachko et al., 2010; Zaharia et al., 2016; Vora, 2011). Some HDFS Big RDF frameworks delegate query processing to centralized RDF stores like RDF-3x (Neumann and Weikum, 2010), offering flexibility and scalability for large RDF datasets. These storage schemes can be broadly classified into (Chawla et al., 2020): (i) *Triple table* (use a single table with subject, predicate, and object columns for RDF triples but become inefficient with data growth, requiring costly self-joins for queries); (ii) *Binary table* (employing two-column tables for each RDF property, addressing null values and multi-valued properties but it result into slow queries involving multiple properties and insert operations); (iii) *Property table* (store triples in wide horizontal tables with n-ary columns, grouping subjects by common properties making it efficient for star pattern SPARQL queries but susceptible to null values and multi-valued attributes); (iv) *Mixed (property-binary table)* (combining property and binary tables mitigate null and multi-valued attribute issues while reducing necessary joins); (v) *Graph-based* (representing RDF data as a labeled directed graph, offering advantages in visualization, flexibility, and integration); (vi) *Hybrid (Triple-based-Graph-based)* (combining triple and graph-based storage, supporting efficient SPARQL query processing and adaptability to specific dataset and query workload requirements).

When selecting the appropriate RDF storage model for a specific application, practitioners should consider dataset size, query workload, data dynamics, and performance requirements.

4.3. Data Mining and Integration

A compelling domain highlighting the advantages of merging Big Data and Semantic Web technologies is data integration. Specifically, in the work by Boury-Brisset (2013), the fusion of Big Data technologies with a semantic layer of ontological models and semantic-based analysis services is employed to facilitate querying, analytics, text annotation, and information extraction. Espinosa Oliva et al. (2015) leverage Big Data techniques to mine heterogeneous data sources and represent the results in LOD format, promoting interoperability and reusability. Additionally, Bartalesi et al. (2023) combines information extraction techniques with Wikidata disambiguation to create LOD-based story maps on a territory from textual data. Furthermore, Truică et al. (2023) use Spark to automatically recognize and extract domain-specific terms that can be further modeled with OntoLex-FRaC (Chiarcos et al., 2022).

4.4. Query Optimization

In the domain of query optimization, the integration of Big Data and Semantic Web technologies holds significant importance. Konstantopoulos et al. (2016) assert that the integration of these technologies offers the advantage of explicating semantics and cross-linking of the data. Furthermore, it facilitates the creation of a unified endpoint capable of federating numerous distributed SPARQL endpoints, including the seamless incorporation of non-RDF data through Apache Solr (Charalambidis et al., 2015). This concept is further reinforced by the proposal of BigOWLIM (Bishop and Bojanov, 2011), an approach aimed not only at query optimization but also at reasoning on extensive knowledge graphs, now available as Ontotext GraphDB [14]. It is important to highlight the importance of available SPARQL endpoints and the difficulties in optimizing federated queries when dealing with larger datasets (Fernández et al., 2017). To support this, the LOD-a-lot method serves very big triple stores via a single, self-indexed Header-Dictionary-Triples (HDT) file, which can either be queried online or downloaded and used locally. Several Big RDF systems leverage Hadoop MapReduce and related Big Data frameworks to optimize and coordinate query processing across distributed clusters of nodes (Chawla et al., 2020; Janev et al., 2020a). Consequently, many joint Big LOD query optimization approaches can be adapted and extended for Big LLOD processing in the context of Big Data and KGs.

5. Processing LLOD using Big Data

In this section, we explore the advantages of utilizing Big Data tools for processing the vast LLOD cloud. By employing these tools, researchers and developers can efficiently manage, process, and analyze large volumes of LOD, thereby gaining valuable insights.

5.1. Big Data Platform: Apache Spark

Apache Spark (Zaharia et al., 2016) is a widely recognized open-source Big Data processing framework that offers fast, scalable, and fault-tolerant data processing capabilities and depicted in Fig. 1. Its in-memory processing engine, coupled with an extensive set of libraries and APIs, has made it a popular choice for handling large-scale data processing tasks across various industries and research domains.

The architecture of Apache Spark is based on a master/worker paradigm, where a driver program manages multiple worker nodes across a distributed computing cluster (Armbrust et al., 2015). The driver program coordinates the execution of tasks across the cluster, manages Resilient Distributed Datasets (RDDs), and communicates with external storage systems and cluster managers. The cluster manager, such as Apache Mesos, Hadoop YARN, or Spark's standalone cluster manager, is responsible for allocating resources like CPU, memory, and network bandwidth to Spark applications. Executors run tasks on worker nodes, manage data storage and caching for RDDs, and report the status of tasks back to the driver program.

The popularity of Apache Spark is due to its versatility, performance, and ease of use. Additionally, it offers a comprehensive set of libraries that cater to a wide range of data processing and analysis tasks, including Spark SQL, Spark Streaming, MLlib, and GraphX.

By leveraging Spark's capabilities, users can effectively process and analyze large-scale data, including data in the LLOD cloud, to extract insights and make data-driven decisions. In a comprehensive benchmarking study (Ragab et al., 2019, 2020, 2021a,b), Apache Spark SQL demonstrated superior performance over Apache Jena in querying large-scale RDF datasets. Specifically, Spark SQL executed queries up to four times faster and used up to 60% less memory on datasets as large as 91 GB. However, Jena was more efficient for smaller datasets and complex queries. The authors suggest Spark SQL as a promising solution for large-scale RDF querying but advocate for additional research to improve its efficiency for intricate operations.

5.2. Big Data Stream Analysis

Big Data batch processing methods are inadequate for analyzing real-time application scenarios, as they cannot handle the demands of instantaneous data analysis. Stream computing, on the other hand, addresses the need for real-time processing of massive, high-velocity data from various sources with minimal latency. In-stream computing, the assumption is that the data's value is intrinsically tied to its freshness, prompting immediate analysis upon arrival in a stream rather than being stored for later analysis as in batch computing. This necessitates the development of parallel architectures and scalable computing platforms, enabling organizations to analyze and respond to rapidly changing data in real-time (Inoubli et al., 2018).

One important application of Big Data stream processing in the fields of linguistics and NLP is real-time event detection in news and social media streams. Numerous studies have employed Spark

[14] https://www.ontotext.com/products/graphdb/

Streaming to identify events on social media platforms (Balachandrudu, 2021), analyze tweet sentiment (Zaki et al., 2020; Patil et al., 2022), and detect instances of hate speech (Doan et al., 2022).

To process the LLOD streaming data, we can employ two approaches. The first one is to use some of the RDF Stream Processing platforms like Continuous SPARQL (C-SPARQL) (Barbieri et al., 2009) or Continuous Query Evaluation over Linked Stream (CQELS) (Le-Tuan et al., 2022). The second approach is to use general-purpose streaming platforms like Spark Streaming (Zaharia et al., 2012) or Apache Kafka (Garg, 2013).

5.3. Distributed Machine Learning

Distributed ML systems can be classified into two main categories: data-parallel and model-parallel (Janbi et al., 2023). In data parallelism, the training data is partitioned across the machines, and each machine computes the gradients on its local data subset. The gradients are then aggregated across the machines to update the model parameters. In model parallelism, the model itself is partitioned across the machines, and each machine computes the gradients on its local model subset. The gradients are then communicated across the machines to update the global model.

There are several tools, frameworks, and libraries that support parallel and distributed processing to speed up model training and inference (Janbi et al., 2023). Several well-known frameworks and libraries, such as TensorFlow (Abadi et al., 2016), PyTorch (Li et al., 2020), and MXNet (Chen et al., 2015), support distributed training in a range of hardware configurations, from single GPUs to clusters of interconnected machines. Although each framework provides different training options, strategies, and paradigms, they all support data parallelism (Janbi et al., 2023). In addition, TensorFlow supports both synchronous and asynchronous training and offers various distribution strategies depending on the underlying hardware (Abadi et al., 2016). PyTorch supports data parallelism as well as other training paradigms, such as pipeline parallelism (Li et al., 2020). MXNet enables data parallelism across multiple machines but only supports model parallelism within a single machine (Chen et al., 2015).

Distributed ML is of essential importance for LLMs. LLMs have recently achieved breakthroughs in NLP tasks, such as language translation, sentiment analysis, and text classification (Liu et al., 2023). However, LLMs require significant computational resources and can take weeks or even months to train on a single machine (Narayanan et al., 2021).

6. Discussion

In this position paper, we argue that Linguistic Linked Open Data and some of its use cases show most of the characteristic aspects of Big Data, i.e. *volume*, *velocity*, *variety*, and *value*. Hence, Big Data techniques may be of use in the LLOD context. This argument draws on the fact that general LOD has already embraced such techniques. However, *Linguistic* LOD exhibits specific aspects that may be even more challenging.

LLOD is usually produced by a myriad of different actors, e.g., corpus linguists, lexicographers, and wiki communities, usually dealing with one or a few languages at a time. This leads to a very scattered data cloud where federated queries have to be used in use cases involving the cloud as a whole.

Also, such data is hybrid in nature, combining highly structured graph-based data with nodes containing language strings where the information is not explicitly structured, e.g., definitions or examples in dictionaries, complex text segments in annotated corpora, or even images. This aspect favours Deep Learning techniques as a good candidate to tackle all the graph and text based information in a common model. This implies a huge need for computing power in order to train embeddings in contexts where velocity is an issue and to handle graph queries along with vector space operators.

The integration of Big Data tools with LLOD offers numerous benefits, including:

- Large-scale data processing: Apache Spark is designed to handle large-scale data processing and can scale horizontally by adding more nodes to the cluster. This makes it well-suited for managing and processing large volumes of LOD.
- Complex data processing: Apache Spark can be used to perform complex data processing tasks, such as data transformations, machine learning, and graph processing. These tasks can be applied to LOD to extract insights or to perform data analysis.
- Integration with other Big Data tools: Apache Spark can be used together with many other Big Data tools like Hadoop and Flink to create a comprehensive Big Data processing stack for LOD
- Fault tolerance: Apache Spark provides built-in fault tolerance so that the data will be always available, even in the event of hardware or software failures.
- Parallel processing: As a distributed processing framework, Apache Spark can perform parallel processing on LOD, which can help to reduce processing time and improve performance
- Stream processing: By employing streaming techniques, it will become feasible to handle the

continuous influx of data, ensuring real-time updates and analyses. Platforms such as Apache Spark, Kafka, Flink, and Storm are well-suited for this purpose.

All three presented use cases can greatly benefit from Big Data techniques, especially the Big Data streaming capabilities, and Big Data machine learning techniques. For the first use case, Big Data techniques can facilitate the dynamic expansion and enhancement of DBnary. These approaches will enable frequent and near-real-time updates of the DBnary dataset. Streaming data processing frameworks, such as Apache Spark or Apache Storm, will allow for the real-time processing of new Wiktionary dumps. Furthermore, machine learning algorithms supported by Apache Spark can be applied to disambiguate translation pairs, thereby enhancing the accuracy of linking word senses across languages. In the second use case, focusing on accessing corpora and linguistic information, Apache Spark can be utilized to analyze extensive corpora over time. This will enable fast and efficient tracking of semantic changes and understanding of language and grammar evolution through large datasets. In the Information Extraction use case, Big Data techniques are indispensable for managing and analyzing the continuous influx of textual data from various sources, such as news articles. Stream processing engines like Apache Kafka and Apache Spark Streaming can efficiently facilitate the dynamic processing and extraction of valuable information from the textual content, including identifying and linking named entities across languages. Machine learning models trained for real-time Information Extraction tasks, such as entity recognition, sentiment analysis, and event detection, can be updated in real time using incremental learning techniques.

The union of LLOD and Big Data could also offer new perspectives to machine learning by facilitating the application of neural approaches to very large-scale Knowledge Graphs and neural approaches, e.g. in the form of Linguistic Graph Neural Networks or knowledge graph infusion to enhance the factual and multilingual knowledge in large language models. A concrete example where the application of Big Data techniques holds great potential for LLOD is link discovery, whereby federated SPARQL queries are replaced with Big Data techniques. This could bring unprecedented efficiency to the solution of well-known problems that include finding translation equivalents, acquiring lexicons for low-resource languages, and extracting information cross-lingually. By providing a fast and efficient platform for exploring LLOD resources that offer a unified, formalized (machine accessible) connection to a wide variety of linguistic resources, the incorporation of Big Data techniques could also help to advance the progress made in such complex areas of linguistic investigation as analysis of diachronic change within and across languages.

One potential risk of this union of Big Data and LLOD we see is that applications of Big Data techniques might be slightly more complicated than LLOD on its own, and solutions should not become so complex that they are not viable. Furthermore, the union requires staying up to date with developments in two fields and having expertise in two fields. Another challenge is that the large collection of language data across languages, description levels, e.g. phonology and semantics, and types of language resources, e.g. corpora and terminologies, need to be collected to hold potential for training language models or other applications. If we collect all these language data, we obtain large, high-quality datasets. However, there is a general lack of computational power and infrastructure, for which the distributed architecture of Big Data provides a solution. Furthermore, LLOD are fragmented and distributed with SPARQL endpoints or as data dumps, which also requires a distributed architecture to collect all this data and run single reliable processes on all of them at once.

Currently, scalability (volume), speed of access for sampling (velocity), and correctness of information (veracity) are well-known issues, however, these topics merit discussion in more detail than space available here permits. Although it is unclear exactly which role, if any, Big Data techniques and frameworks might play, the higher the number of languages and the greater the variety of data in a knowledge graph, the more pertinent these issues become.

7. Conclusion

In this position paper, we argue that if we want to benefit from the potential of the LLOD cloud to become a directly accessible very large dataset of high-quality data, we need to move from triple stores, data dumps, and federated queries to SPARQL endpoints to processing the LLOD with Big Data techniques. The distributed architecture holds the potential to access and process fragmented and distributed LLOD resources at once. We specify and exemplify this potential in form of concrete use cases, which are uncovering translation mappings across languages, accessing linguistic information, and extracting information across languages. To foster this union of LLOD and Big Data, the first steps will be to provide training events so that experts in one field can acquire knowledge on the other, and networking meetings to exchange ideas and expertise.

8. Acknowledgements

This paper was based upon work in COST Action CA18209 Nexus Linguarum, supported by COST (European Cooperation in Science and Technology). http://www.cost.eu/. C.O. Truică and E.S. Apostol are supported in part by project DECIP ("Dezvoltarea Capacitatii Institutionale a Universitatii POLITEHNICA din Bucuresti", contract no. PFE57/2022) and by the National University of Science and Technology Politehnica Bucharest through the PubArt program.

9. Bibliographical References

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Gregory S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian J. Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Józefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Rajat Monga, Sherry Moore, Derek Gordon Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul A. Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda B. Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2016. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *CoRR*, abs/1603.04467.

Hemn Barzan Abdalla. 2022. A brief survey on big data: technologies, terminologies and data-intensive applications. *Journal of Big Data*, 9(1):1–36.

Florentina Armaselu, Elena-Simona Apostol, Anas Fahad Khan, Chaya Liebeskind, Barbara McGillivray, Ciprian-Octavian Truică, Andrius Utka, Giedrė Valūnaitė Oleškevičienė, and Marieke van Erp. 2022. LL(O)D and NLP perspectives on semantic change for humanities research. *Semantic Web*, 13(6):1051–1080.

Michael Armbrust, Reynold S. Xin, Cheng Lian, Yin Huai, Davies Liu, Joseph K. Bradley, Xiangrui Meng, Tomer Kaftan, Michael J. Franklin, Ali Ghodsi, and Matei Zaharia. 2015. Spark SQL: relational data processing in spark. In *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data, Melbourne, Victoria, Australia, May 31 - June 4, 2015*, pages 1383–1394. ACM.

Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. 2007. Dbpedia: A nucleus for a web of open data. In *The Semantic Web*, pages 722–735, Berlin, Heidelberg. Springer Berlin Heidelberg.

KE Balachandrudu. 2021. Identifying events in social media streams in real time using semantic analysis of irrelevant phrases. *Webology (ISSN: 1735-188X)*, 18(4).

Davide Francesco Barbieri, Daniele Braga, Stefano Ceri, Emanuele Della Valle, and Michael Grossniklaus. 2009. C-SPARQL: SPARQL for continuous querying. In *Proceedings of the 18th International Conference on World Wide Web, WWW 2009, Madrid, Spain, April 20-24, 2009*, pages 1061–1062. ACM.

Valentina Bartalesi, Gianpaolo Coro, Emanuele Lenzi, Pasquale Pagano, and Nicolò Pratelli. 2023. From unstructured texts to semantic story maps. *International Journal of Digital Earth*, 16(1):234–250.

Barry Bishop and Spas Bojanov. 2011. Implementing OWL 2 RL and OWL 2 QL rule-sets for OWLIM. In *Proceedings of the 8th International Workshop on OWL: Experiences and Directions (OWLED 2011), San Francisco, California, USA, June 5-6, 2011*, volume 796 of *CEUR Workshop Proceedings*. CEUR-WS.org.

Christian Bizer, Tom Heath, and Tim Berners-Lee. 2009. Linked data - the story so far. *International Journal on Semantic Web and Information Systems (IJSWIS)*, 5(3):1–22.

Anne-Claire Boury-Brisset. 2013. Managing Semantic Big Data for Intelligence. In *STIDS 2013 Semantic Technologies for Intelligence, Defense, and Security*, volume 1097, pages 41–47.

Angelos Charalambidis, Antonis Troumpoukis, and Stasinos Konstantopoulos. 2015. Semagrow: optimizing federated SPARQL queries. In *Proceedings of the 11th International Conference on Semantic Systems, SEMANTiCS 2015, Vienna, Austria, September 15-17, 2015*, pages 121–128. ACM.

Tanvi Chawla, Girdhari Singh, and Emmanuel S. Pilli. 2021. MuSe: a multi-level storage scheme for big RDF data using MapReduce. *Journal of Big Data*, 8(1):130.

Tanvi Chawla, Girdhari Singh, Emmanuel S. Pilli, and M.C. Govil. 2020. Storage, partitioning, indexing and retrieval in big rdf frameworks: A survey. *Computer Science Review*, 38:100309.

Tianqi Chen, Mu Li, Yutian Li, Min Lin, Naiyan Wang, Minjie Wang, Tianjun Xiao, Bing Xu, Chiyuan Zhang, and Zheng Zhang. 2015. MXNet: A flexible and efficient machine learning library for heterogeneous distributed systems. *CoRR*, abs/1512.01274.

Christian Chiarcos, Elena-Simona Apostol, Besim Kabashi, and Ciprian-Octavian Truică. 2022. Modelling frequency, attestation, and corpus-based information with OntoLex-FrAC. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 4018–4027, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Christian Chiarcos, Sebastian Hellmann, and Sebastian Nordhoff. 2011. Towards a Linguistic Linked Open Data cloud: The Open Linguistics Working Group. *Traitement Automatique des Langues*, 52:245–275.

Christian Chiarcos, Sebastian Hellmann, Sebastian Nordhoff, Steven Moran, Richard Littauer, Judith Eckle-Kohler, Iryna Gurevych, Silvana Hartmann, Michael Matuschek, and Christian M. Meyer. 2012. The open linguistics working group. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 3603–3610, Istanbul, Turkey. European Language Resources Association (ELRA).

Christian Chiarcos and Gilles Sérasset. 2022. A cheap and dirty cross-lingual linking service in the cloud. In *Proceedings of the 8th Workshop on Linked Data in Linguistics within the 13th Language Resources and Evaluation Conference*, pages 52–60, Marseille, France. European Language Resources Association.

Richard Cyganiak, David Wood, and Markus Lanthaler. 2014. RDF 1.1 Concepts and Abstract Syntax. W3C Recommendation 25 February 2014.

Long-An Doan, Phuong-Thao Nguyen, Thi-Oanh Phan, and Trong-Hop Do. 2022. An implementation of large scale hate speech detection system for streaming social media data. In *2022 IEEE International Conference on Communication, Networks and Satellite (COMNETSAT)*, pages 155–159.

Roberto Espinosa Oliva, Larisa Garriga, Jose Zubcoff, and Jose-Norberto Mazon. 2015. Linked open data mining for democratization of big data. *Proceedings - 2014 IEEE International Conference on Big Data, IEEE Big Data 2014*, pages 17–19.

Marziye Fathi, Mostafa Haghi Kashani, Seyed Mahdi Jameii, and Ebrahim Mahdipour. 2021. Big data analytics in weather forecasting: A systematic review. *Archives of Computational Methods in Engineering*, 29.

Javier D. Fernández, Wouter Beek, Miguel A. Martínez-Prieto, and Mario Arias. 2017. Lod-a-lot. In *The Semantic Web – ISWC 2017*, pages 75–83, Cham. Springer International Publishing.

Nishant Garg. 2013. *Apache kafka*. Packt Publishing Birmingham, UK.

Jorge Gracia, Marta Villegas, Asuncion Gomez-Perez, and Nuria Bel. 2018. The apertium bilingual dictionaries on the web of data. *Semantic Web*, 9(2):231–240.

William L. Hamilton, Jure Leskovec, and Dan Jurafsky. 2016. Diachronic word embeddings reveal statistical laws of semantic change. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1489–1501, Berlin, Germany. Association for Computational Linguistics.

Herodotos Herodotou, Despoina Chatzakou, and Nicolas Kourtellis. 2020. A streaming machine learning framework for online aggression detection on twitter. In *2020 IEEE International Conference on Big Data (Big Data)*, pages 5056–5067.

Wissem Inoubli, Sabeur Aridhi, Haithem Mezni, Mondher Maddouri, and Engelbert Mephu Nguifo. 2018. A comparative study on streaming frameworks for Big Data. In *LADaS 2018 Latin America Data Science Workshop*, volume 2170, pages 17–24. CEUR Workshop Proceedings.

Nourah Janbi, Iyad Katib, and Rashid Mehmood. 2023. Distributed artificial intelligence: Review, taxonomy, framework, and reference architecture. *Taxonomy, Framework, and Reference Architecture (January 1, 2023)*.

Valentina Janev, Damien Graux, Hajira Jabeen, and Emanuel Sallinger. 2020a. *Knowledge graphs and big data processing*. Lecture Notes in Computer Science. Springer International Publishing, Chan.

Valentina Janev, Dea Pujić, Marko Jelić, and Maria-Esther Vidal. 2020b. Chapter 9 survey on big data applications. In Valentina Janev, Damien Graux, Hajira Jabeen, and Emanuel Sallinger, editors, *Knowledge Graphs and Big Data Processing*, pages 149–164. Springer International Publishing, Cham.

Marijn Janssen and Jeroen van den Hoven. 2015. Big and open linked data (bold) in government: A challenge to transparency and privacy? *Government Information Quarterly*, 32(4):363–368.

Stasinos Konstantopoulos, Angelos Charalambidis, Giannis Mouchakis, Antonis Troumpoukis, Jürgen Jakobitsch, and Vangelis Karkaletsis. 2016. Semantic Web Technologies and Big Data Infrastructures: SPARQL Federated Querying of Heterogeneous Big Data Stores. In *Proceedings of the ISWC 2016 Posters & Demonstrations Track*, volume 1690. CEUR Workshop Proceedings.

Anh Le-Tuan, Manh Nguyen Duc, Chien-Quang Le, Trung-Kien Tran, Manfred Hauswirth, Thomas Eiter, and Danh Le Phuoc. 2022. CQELS 2.0: Towards A unified framework for semantic stream fusion. *CoRR*, abs/2202.13958.

Shen Li, Yanli Zhao, Rohan Varma, Omkar Salpekar, Pieter Noordhuis, Teng Li, Adam Paszke, Jeff Smith, Brian Vaughan, Pritam Damania, et al. 2020. PyTorch Distributed: Experiences on accelerating data parallel training. *PVLDB*, 13(12).

Ying Li and Cynthia SQ Siew. 2022. Diachronic semantic change in language is constrained by how people use and learn language. *Memory & Cognition*, 50(6):1284–1298.

Yiheng Liu, Tianle Han, Siyuan Ma, Jiayue Zhang, Yuanyuan Yang, Jiaming Tian, Hao He, Antong Li, Mengshen He, Zhengliang Liu, Zihao Wu, Dajiang Zhu, Xiang Li, Ning Qiang, Dinggang Shen, Tianming Liu, and Bao Ge. 2023. Summary of chatgpt/gpt-4 research and perspective towards the future of large language models. *CoRR*, abs/2304.01852.

John McCrae, Dennis Spohr, and Philipp Cimiano. 2011. Linking lexical resources and ontologies on the semantic web with lemon. In *The Semantic Web: Research and Applications*, pages 245–259, Berlin, Heidelberg. Springer.

John P McCrae, Julia Bosque-Gil, Jorge Gracia, Paul Buitelaar, and Philipp Cimiano. 2017. The ontolex-lemon model: development and applications. In *Electronic lexicography in the 21st century. Proceedings of eLex 2017 conference.*, pages 19–21. Lexical Computing CZ s.r.o.

JP McCrae and T Declerck. 2019. Linguistic linked open data for all. *Proceedings of Language Technology 4 All*, pages 13–15.

Jean-Baptiste Michel, Yuan Kui Shen, Aviva Presser Aiden, Adrian Veres, Matthew K Gray, Google Books Team, Joseph P Pickett, Dale Hoiberg, Dan Clancy, Peter Norvig, et al. 2011. Quantitative analysis of culture using millions of digitized books. *Science*, 331(6014):176–182.

Damir Mukhamedshin, Olga Nevzorova, and Alexander Kirillovich. 2020. Using floss for storing, processing and linking corpus data. In *Open Source Systems*, pages 177–182, Cham. Springer International Publishing.

Muhammad Naeem, Tauseef Jamal, Jorge Diaz-Martinez, Shariq Aziz Butt, Nicolo Montesano, Muhammad Imran Tariq, Emiro De-la Hoz-Franco, and Ethel De-La-Hoz-Valdiris. 2022. Trends and future perspective challenges in big data. In *Advances in Intelligent Data Analysis and Applications*, pages 309–325, Singapore. Springer Singapore.

Deepak Narayanan, Mohammad Shoeybi, Jared Casper, Patrick LeGresley, Mostofa Patwary, Vijay Korthikanti, Dmitri Vainbrand, Prethvi Kashinkunti, Julie Bernauer, Bryan Catanzaro, Amar Phanishayee, and Matei Zaharia. 2021. Efficient large-scale language model training on GPU clusters using megatron-lm. In *International Conference for High Performance Computing, Networking, Storage and Analysis, SC 2021, St. Louis, Missouri, USA, November 14-19, 2021*. ACM.

Thomas Neumann and Gerhard Weikum. 2010. The rdf-3x engine for scalable management of rdf data. *The VLDB Journal*, 19:91–113.

Giedrė Valūnaitė Oleškevičienė, Chaya Liebeskind, Dimitar Trajanov, Purificação Silvano, Christian Chiarcos, and Mariana Damova. 2021. Speaker attitudes detection through discourse markers analysis. *Deep Learning and Neural Approaches for Linguistic Data*, page 8.

Rahul Patil, Swapnil Harwalkar, Kaustubh Ingale, Vishwajeet Patil, and Soham Puranik. 2022. Mining social media data streams for sentimental analysis. *International Journal of Engineering Research & Technology (IJERT)*, 11.

Sancheng Peng, Guojun Wang, and Dongqing Xie. 2017. Social influence analysis in social networking big data: Opportunities and challenges. *IEEE Network*, 31(1):11–17.

Bruno Pouliquen, Marco Kimler, Ralf Steinberger, Camelia Ignat, Tamara Oellinger, Ken Blackler, Flavio Fluart, Wajdi Zaghouani, Anna Widiger, Ann-Charlotte Forslund, and Clive Best. 2006. Geocoding multilingual texts: Recognition, disambiguation and visualisation. In *Proceedings* of the Fifth International Conference on Language Resources and Evaluation (LREC'06)*, Genoa, Italy. European Language Resources Association (ELRA).

Mohamed Ragab, Feras M. Awaysheh, and Riccardo Tommasini. 2021a. Bench-Ranking: A first step towards prescriptive performance analyses for big data frameworks. In *2021 IEEE International Conference on Big Data (Big Data), Orlando, FL, USA, December 15-18, 2021*, pages 241–251. IEEE.

Mohamed Ragab, Riccardo Tommasini, Feras M. Awaysheh, and Juan Carlos Ramos. 2021b. An in-depth investigation of large-scale RDF relational schema optimizations using spark-sql. In *Proceedings of the 23rd International Workshop on Design, Optimization, Languages and Analytical Processing of Big Data (DOLAP) co-located with the 24th International Conference on Extending Database Technology and the 24th International Conference on Database Theory (EDBT/ICDT 2021), Nicosia, Cyprus, March 23, 2021*, volume 2840, pages 71–80. CEUR Workshop Proceedings.

Mohamed Ragab, Riccardo Tommasini, Sadiq Eyvazov, and Sherif Sakr. 2020. Towards making sense of spark-sql performance for processing vast distributed RDF datasets. In *Proceedings of The International Workshop on Semantic Big Data, SBDSIGMOD 2020, Portland, Oregon, USA, June 19, 2020*, pages 1:1–1:6. ACM.

Mohamed Ragab, Riccardo Tommasini, and Sherif Sakr. 2019. Benchmarking Spark-SQL under alliterative RDF relational storage backends. In *Proceedings of the QuWeDa 2019: 3rd Workshop on Querying and Benchmarking the Web of Data co-located with 18th International Semantic Web Conference (ISWC 2019), Auckland, New Zealand, October 26-30, 2019*, volume 2496 of *CEUR Workshop Proceedings*, pages 67–82. CEUR-WS.org.

Gilles Sérasset. 2015. DBnary: Wiktionary as a lemon-based multilingual lexical resource in RDF. *Semantic Web*, 6(4):355–361.

Konstantin Shvachko, Hairong Kuang, Sanjay Radia, and Robert Chansler. 2010. The hadoop distributed file system. In *IEEE 26th Symposium on Mass Storage Systems and Technologies, MSST 2012, Lake Tahoe, Nevada, USA, May 3-7, 2010*, pages 1–10. IEEE Computer Society.

Miguel-Angel Sicilia, Salvador Sánchez-Alonso, and Elena Barriocanal. 2016. Sharing linked open data over peer-to-peer distributed file systems: The case of ipfs. In *Metadata and Semantics Research: 10th International Conference, MTSR 2016, Göttingen, Germany, November 22-25, 2016, Proceedings*, pages 3–14. Springer.

Andon Tchechmedjiev, Gilles Sérasset, Jérôme Goulian, and Didier Schwab. 2014. Attaching Translations to Proper Lexical Senses in DBnary. In *3rd Workshop on Linked Data in Linguistics: Multilingual Knowledge Resources and Natural Language Processing*, Reykjavik, Iceland.

Ciprian-Octavian Truică, Florin Rădulescu, Alexandru Boicea, and Ion Bucur. 2015. Performance evaluation for CRUD operations in asynchronously replicated document oriented database. In *2015 20th International Conference on Control Systems and Computer Science*, pages 191–196. IEEE.

Ciprian-Octavian Truică and Elena-Simona Apostol. 2021. NoSQL environments and big data analytics for time series. In *Data Science and Big Data Analytics in Smart Environments*, pages 108–138. CRC Press.

Ciprian-Octavian Truică, Neculai-Ovidiu Istrate, and Elena-Simona Apostol. 2023. A distributed automatic domain-specific multi-word term recognition architecture using spark ecosystem. In *The 22nd IEEE International Symposium On Parallel And Distributed Computing (ISPDC2023)*, pages 31–38.

Mehul Nalin Vora. 2011. Hadoop-hbase for large-scale data. In *Proceedings of 2011 International Conference on Computer Science and Network Technology*, volume 1, pages 601–605. IEEE.

Matei Zaharia, Tathagata Das, Haoyuan Li, Scott Shenker, and Ion Stoica. 2012. Discretized streams: An efficient and fault-tolerant model for stream processing on large clusters. In *4th USENIX Workshop on Hot Topics in Cloud Computing, HotCloud'12, Boston, MA, USA, June 12-13, 2012*. USENIX Association.

Matei Zaharia, Reynold S. Xin, Patrick Wendell, Tathagata Das, Michael Armbrust, Ankur Dave, Xiangrui Meng, Josh Rosen, Shivaram Venkataraman, Michael J. Franklin, Ali Ghodsi, Joseph Gonzalez, Scott Shenker, and Ion Stoica. 2016. Apache Spark: a unified engine for big data processing. *Commun. ACM*, 59(11):56–65.

Nashwan Dheyaa Zaki, Nada Yousif Hashim, Yasmin Makki Mohialden, Mostafa Abdulghafoor Mohammed, Tole Sutikno, and Ahmed Hussein

Ali. 2020. A real-time big data sentiment analysis for iraqi tweets using spark streaming. *Bulletin of Electrical Engineering and Informatics*, 9(4):1411–1419.

10. Language Resource References

Sérasset, Gilles. 2015. *DBNary*. GETALP-LIG – Université Grenoble Alpes", National Research Council, in Pisa. PID https://kaiko.getalp.org/about-dbnary/.