Connect, Understand and Learn: Dynamic Knowledge Graph Transforms Learning

Marlies Morgen*, Lejla Begic Fazlic* and Guido Dartmann*
*Institute for Software Systems (ISS), Trier University of Applied Sciences,
Birkenfeld, Germany

***Abstract*—The automation of knowledge graphs is a challenge if only small training data sets are available for the corresponding learning methods. The approach presented in this paper can work with small training data sets and enables the solution of tasks with previously hidden syntactic structures. In this research, a new conceptual algorithm for learning and updating knowledge graphs is proposed. We combined a powerful NLP approach with statistical methods to build a word frequency-based corpus for various question answering problems. Then, we used specific similarity measures to find the best answer for the given problem. For this purpose, a vector model is used and weights are calculated for the association between terms and problems. In the last phase, we created a continuous learning model with a dynamic knowledge graph that can be updated with new tasks and predict answers to upcoming problems. The knowledge graph is updated with new information when the pattern of a problem is unknown and therefore not found. The implementation of the algorithm is validated using various openly available data sets from the field of user support in business and medicine. The proposed method supports an incremental learning approach and real-time implementation.**

***Index Terms*—Data Mining, Natural Language Processing, Machine Learning, TF-IDF, Knowledge Graph**

I. Introduction

The amount of information on the web is constantly expanding. The goal of the Semantic Web is to use software agents to perform autonomous information integration for a wide range of resources. It contains a large amount of information in the form of knowledge graphs (KGs), where answering questions is recognized as an efficient but nontrivial task [1]. Ontology in Semantic Web helps to integrate data, for example, when ambiguities in expressions are used in different data sets or when new connections between them can be discovered with very little knowledge. Over the past year, KGs have been a critical component of finding information on popular search engines e.g. Google[1] etc. Giving the answer for a problem description that involves multiple entities and relations is a challenging task in text based KGs. This problem is particularly evident if the answer contains the combination of two or even more answers. Ontology construction along with data annotation and relation extraction, as an indispensable factor in KG construction process is not a trivial process in data acquisition. The goal in our research is to develop a conceptual algorithm design for KG learning, prediction and adaptation. To achieve this goal, we have used the Natural Language Processing (NLP) along with the statistic approach Term Frequency - Inverse Document Frequency (TF-IDF) and data mining. The developed design dynamically updates the KG with upcoming information and is able to make predictions about potential answers based on a similarity threshold. Our study additionally investigates energy characteristics of our approach that involves dynamic KG creation and visualization, segmented by data sizes to understand their impact on energy consumption and performance.

Section II reviews recent studies in dynamic KG construction and question-answer (QA) systems. Section III describes statistical and learning methods used in this work. Sections IV and V present our proposed model, especially the NLP-TF-IDF system design architecture and the framework Rasa as the user interface. Section VI addresses the resource efficiency of our approach. Section VII discusses the project's future directions, and Section VIII summarizes the results and conclusions.

II. State of the Art

In developing our proposed approach, both the challenges of creating and extending KGs for question answering systems and developing the conversational system itself played a crucial role. Consequently, our investigation of the state of the art will be tailored to these key areas.

The authors in [2] proposed an unconstrained multi-hop reasoning network. They addressed two problems in graph-based question mapping: The challenge of one-to-many and one-to-one mapping between questions and relations. The problem of entity semantic linking in context of relation extraction using deep learning, active and transfer learning including supervised, semi-supervised and unsupervised algorithms is discussed in the detailed survey by the authors in [3]. A paper review of methods that attempt to automate ontology construction and the challenges involved are presented by the authors in [4].

To address the challenges of dynamic KGs, the authors of [5] have devised the Dynamic Reasoning Network. The interpretative neural model is used to predict relationships in each step of the reasoning process. Pursuing another strategy, the authors in [6] propose a dynamic graph framework based on deep reinforcement learning. They developed a novel dynamic reward function to acquire the dynamic reasoning model. To update existing KGs,

[1]https://support.google.com/knowledgepanel/answer/9787176?hl=en&sjid=16070655908403787963-EU

researchers in [7] implemented a solution consisting of agents that operate using ontology defined concepts and instances. In this context, the authors in [8] proposed the KEQA framework based on the pre-trained KG embeddings. Since training questions and KG are always updated dynamically, this scenario was not considered in the proposed framework.

Regarding conversational systems, [9] proposes a QA system that delivers the responses to user questions by creating and querying SPARQL queries. There are alternatives to the NLP to SPARQL query conversion we use. One of them is NLP2SQL [10], which converts natural language texts into SQL queries. A supervised iterative approach for the specific subgraph's construction using graph convolutional network (CNN) for sub-graph nodes identification and answer extraction is proposed by the authors of [11]. It enables the model to effectively answer complex queries by pulling together disparate pieces of information. Different learning approach frameworks and novel entity detection models are proposed to resolve coreference in conversation [12], [13].

III. Methodology

We used two different areas to test our approach: the enterprise and the medical domain. Each of the data sets was treated individually by performing specific data extraction and pre-processing. One algorithm was then used throughout to create dynamic KGs for each use case. In the enterprise area, we used a total of three different data sets for training and validation purposes: two open available data sets[2,3] and one proprietary data set collected from QA of various IT support platforms. The customer support data sets, which include data on IT problems, allow us to test our approach against real-world scenarios. For example, if a user makes an inquiry about frequent internet interruptions, the system selects the most relevant troubleshooting recommendations, demonstrating our system's ability to adapt solutions based on user context and query semantics. For the medical domain, we extracted question-answer pair from different medical guidelines [14]. Additionally, we used the Covid-Q open source data set provided by the authors of [15]. These data sets allow us to validate our approach to health-related inquiries. For example, in the case of a request to reduce blood pressure, our system identifies and suggests the most appropriate recommendations from several possible strategies, demonstrating the precision and adaptability of our methodology.

A. Conditional Random Fields

Conditional Random Field (CRF) [16] is used primarily for sequence tagging within our text processing pipeline, crucial for the high contextual accuracy required in medical text analysis. CRFs manage dependencies between tags in sequences, which is vital for maintaining coherent tagging sequences in complex medical information. For instance, CRFs ensure that "high temperature" is tagged as a medical condition, avoiding misinterpretation with non-medical contexts. After CRFs tag the components of the text, Decision Trees classify these tags into refined and precise categories. This classification is essential for integrating the extracted data into a dynamic KG, enhancing the granularity of information. For example, after a CRF tags "ibuprofen" as a medication, a Decision Tree further classifies it as an "anti-inflammatory", based on additional contextual features captured by the CRF. Based on the determination of the order and the determination of a split-point in the sentence described above, we are able to split the sentence into two parts: problem and answer. We used the final result of this step as input data set for algorithm implementation (see code 1).

B. Term Frequency - Inverse Document Frequency

In Natural Language Processing, TF-IDF is a statistical model, usually used to overcome the limitation in the Bag of Word (BOW) methodology [17]. It is a very useful tool in recommendation, classification and information retrieval applications. It is defined as "formal measure of how concentrated into relatively few documents are the occurrences of a given word" [18]. The model works in two phases: In the first phase, Term Frequency (TF) is multiplied with Inverse Document Frequency (IDF) as a local component. In the next phase, the result is normalized to unit length by TF-IDF This methodology gives importance to a term in a document if it occurs more than once in that document, or penalizes it if a term occurs very frequently in different documents in the corpus. It is defined by the following product:

Definition 1. *[18]*

Let N be the total number of documents, let n_i be the number of documents in which the word with index i exist, and Let $f_{i,j}$ be the frequency of a word with index i in a document with index j, then the TF-IDF $TF.IDF$ for a word i in a document j is given by the product

$$TF.IDF = TF_{i,j} \cdot IDF_i \tag{1}$$

where

$$TF_{i,j} = \frac{f_{i,j}}{\max_k f_{k,j}} \tag{2}$$

and

$$IDF_i = \log(\frac{N}{n_i}) \tag{3}$$

C. Dynamic Knowledge Graph

In recent years, knowledge graphs have become very popular as a new way of representing knowledge. There is no single definition of knowledge graphs, but a variety of descriptions and definitions that have been published over the past decade [19]. A general and informal definition has been given by Paulheim in [20]:

[2] https://www.kaggle.com/datasets/utsav15/it-helpdesk
[3] https://www.kaggle.com/datasets/suraj520/customer-support-ticket-dataset

```xml
<?xml version="1.0" encoding="utf-8"?>
<record>
    <input>In case LAN not is working in a VM change network settings to Bridged and
    configure network in VM.</input>
    <condition>In case LAN not is working in a VM</condition>
    <action>change network settings to Bridged and configure network in VM.</action>
</record>
```

Listing 1: Example of an information extraction from medical guideline using Conditional Random Fields (CRF)

Definition 2. *[21] A knowledge graph (i) mainly describes real world entities and their interrelations, organized in a graph, (ii) defines possible classes and relations of entities in a schema, (iii) allows for potentially interrelating arbitrary entities with each other and (iv) covers various topical domains.*

Reasoning via graphs is particularly attractive if new knowledge and conclusions can be gained from the data. Rule-based, distributed and neural-based reasoning via KGs as well as their applications in expert systems and possible challenges are summarized by the authors in [21].

Definition 3. *[22] Let $\mathcal{V}$ be the set of nodes and $\mathcal{E}$ the set of edges and $\mathcal{R} = r_1, r_2, ..., r_R$ be the set of relations then a knowledge graph is a directed graph $G = (\mathcal{V}, \mathcal{E})$ where each node $v_k \in \mathcal{V}$ represents an entity or concept and each edge $e_k = (v_i, v_j)$ represents a relation between two nodes, i.e., $E \subseteq \mathcal{V} \times \mathcal{R} \times \mathcal{V}$. In addition, an edge may have weight or a relation type (e.g., "is part of").*

In our Algorithm, we are using an extension of KG by so called continuous-time dynamic graphs.

Definition 4. *[22] A continuous-time dynamic graph is defined as a pair $(G, \mathcal{S})$ where $\mathcal{S}$ represents the set of observations and graph G represents initial state of the dynamic graph at time t_0. Every observation is defined as tuple of event type, event and timestamp. The event type can be an edge, node addition, edge removal, node merging, etc. In any time point t, a graph G_t can be constructed by sequentially updating G according to the observations $\mathcal{S}$ that occurred before or at time t. A discrete time dynamic graph is defined as set of $(G_1, ..G_t)$, where G_t represents a graph at time t defined as $(N_t, \mathcal{E}_t)$, where N_t and $\mathcal{E}_t$ represent sets of nodes and edges in G_t, respectively.*

Fig. 1 shows the transition of a KG between states t_1 and t_2. The term dynamic knowledge graph refers to both continuous and discrete-time KGs. The first step of our approach was investigating a KG node classification and developing a model for classifying each node into one of pre-defined classes. Then, we considered reasoning over the dynamic graph e.g. the interpolation problem and investigated how to add a new node and how to predict links between the nodes. Through continuous learning, our approach is able to learn independently and adapt a KG when new data arrives.

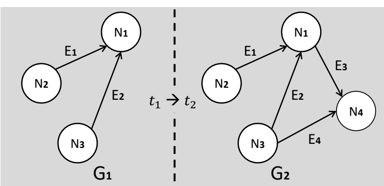

Fig. 1: Dynamic Knowledge Graph Transition

IV. Algorithm Design

We used open-source data sets and a generated data set in form of QA from textual data using CRF. This involved employing CRFs to analyze the grammatical and syntactic structure of sentences, enabling identification and extraction of question-answer formats. By engineering specific features such as word tokens and Part-of-speech tags, we trained the CRF model to recognize and differentiate between questions and answers within the text, effectively creating a structured QA data set from unstructured data sources. For example, the sentence *"In a case of Covid and high temperature, it is allowed to use paracetamol"* contains an answer to the question *"Which medications are commonly used to alleviate the symptoms of Covid?"* or similarly *"What can be used in the event of Covid and high temperature?"*. The answer in both cases is *"Paracetamol"*. We extracted these QA pairs as a new data set for our proposed algorithm by extracting problem (labeled as condition in Lst. 1) and answer (labeled as action in Lst. 1) from a given input. The training and testing (80/20 split) accuracy of the classification [23] for our model was 0.97 and 0.89 respectively.

To calculate the f_1 score [24], we define true-positive data sets as those in which the problem was correctly detected. Accordingly, records are classified as false-positive if they were not fully recognized correctly. The second category also includes cases where the problem was only partially extracted or where the answer is additionally included in the extracted problem. Classification as negative here refers to the extraction of answers. True-negative values are completely correctly recognized answers, while false-negative classified answers were not completely recognized or contain parts of the problem. The overall f_1 score for the entire data set is approximately 0.93.

An important step in the manipulation of the text model is data cleaning and pre-processing (1 in Fig. 2). In the first step of the document cleaning phase, we removed all unnecessary symbols, punctuation marks, and duplicate or

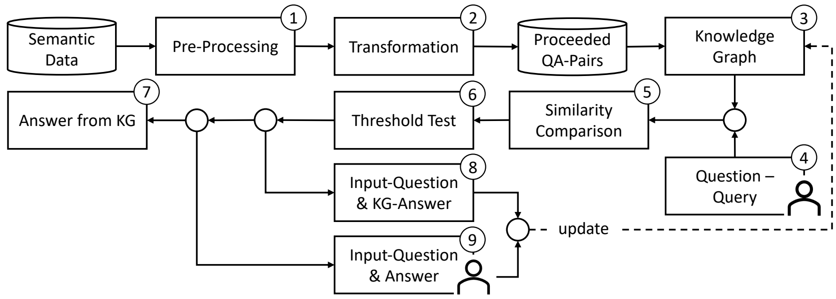

Fig. 2: Algorithm design

missing data. Using NLP techniques, we removed stop words and lower-cased each token of the input text.

After using NLP to tokenize the words in the problem description, we create a BOW dictionary. This representation contains the word identification (ID) and its frequency in the document. BOW does not consider the relevance of words across documents, so we used the TF-IDF (see Definition 1) approach (2 in Fig. 2) to enhance the effectiveness of information retrieval. In this way, we created a model with a corpus and the cosine similarity measure. In this method, a term is given importance in a document if it occurs more than once in that document, or it is penalized if it occurs very frequently in different documents in the corpus. TF-IDF ignores frequent words and gives more importance to rare words. In this phase, we performed a similarity query against the corpus.

In the second step, using the information results from the previous step, a KG is initialized and constructed using the NetworkX/Matplotlib library [25], [26] (3 in Fig. 2). We created a KG for an existing (offline) model using the rdflib library [27]. Our algorithm learns templates created from QA pairs, following the approach from [28]. The input QA pairs consist of question utterance u and the corresponding answer set A_u from the knowledge graph. An example of a training utterance is "Login credentials do not work." which is paired with the answer set A_u = "Go to URL: reset.com, Enter you login id, enter password received, reset password". We model structured information about extracted entities and relations in terms of triples (subject - predicate - object). This way we can efficiently retrieve detailed information by navigating through the network of data points, often visualized as a graph. The process involves converting textual data into numerical representations, which is a common step in machine learning models. Furthermore, to effectively interpret and respond to queries, the model relies on a vocabulary that is constructed from the training data. This vocabulary serves as base to understand and generate language, allowing the model to recognize patterns, context, and semantics in the text it processes. We converted the text to numbers and created a vocabulary. Using TF-IDF, we calculated how many times each word occurs in the given data set to obtain a numerical overview of the importance of the words in the query for statistical analysis. The next step is related to the processing of user queries. We investigated if an existing query of the KG can be used to answer a query and if it is necessary to update the KG with the input to extend it. For a new query (4 in Fig. 2), we used cosine similarity to check the similarity between the user query and existing queries in the graph (5 in Fig. 2). Thus, we find the existing query in the knowledge graph that has the highest similarity score and compare it with the new query (6 in Fig. 2). When the cosine similarity threshold of 0.4 is exceeded, the user is offered the answer to the existing query with the highest similarity value (7 in Fig. 2). Here we have analyzed two different cases: If A.) a query does not match any node of the knowledge graph (does not exist in the knowledge graph) or B.) is similar but not equal (similarity score is under 98%) to an existing query, we add the new query as node in the KG and connect it with high scoring answers (8 in Fig. 2). In case of A.) user feedback is used to update the knowledge graph with a new QA pair (9 in Fig. 2). This way the model is continuously learning and adapting on its own as new data requests come in. The implementation of the algorithm in this research paper has been validated against several openly available data sets from the customer support and medical domains. The proposed method supports an incremental learning approach and real-time implementation.

V. Rasa

Rasa Technologies Inc. developed *Rasa*[4], a framework proving functions to build chat and voice-based AI assistants. We created rules, stories and Natural Language Understanding (NLU) training data to build a scenario based chatbot. We implemented functions for various use cases in which users communicate with the chatbot:

- Users can extend the data set with additional pairs consisting of problem and matching answer.
- Users can create and update the knowledge graph based on the data set.
- For customer queries, we focused on question-answer pairs. If users enter an issue, they receive the most appropriate answers as output.
- Users can retrieve information about customers, employees and tickets. This includes analyses such as response times, ticket data and contact information.

[4]https://rasa.com/docs/rasa/

We integrated the prepared code blocks for Rasa [5] into the provided structure and made some style adjustments. The input is a question, e.g. on an IT topic, to the chatbot in the user interface, which is processed in the back-end, and then the user is presented with a list of possible answers to the problem. The order within this list of results is determined by the highest match, so the user is offered the most likely answer first.

VI. Energy Efficiency Analysis

We examined the energy consumption characteristics of the proposed approach in the domain of customer support data sets (Tab. I). This examination primarily focused on two key actions: dynamic knowledge graph learning, which involves the construction and organization of interconnected data nodes to represent relationships within the data set, and visualization, which entails the graphical representation of the created knowledge graph to facilitate the interpretation and analysis of the data set. Execution time, energy consumption and CPU were investigated as key performance metrics across different data sizes in the energy measurements analysis. The measurement method and experiment setup adhere to the methodology and guidelines outlined in the Green Software Measurement Model by the authors in [29]. Measurements were conducted on a computer equipped with 4GB of RAM (configured as two 2GB modules), powered by an Intel Core i5-650 processor, and featuring a dual-storage setup with a 500GB HDD and a 250GB SSD, providing a blend of ample storage and efficient data access. In this study, we assessed and evaluated the energy requirements across three distinct scenarios throughout the KG life cycle. Certain energy consumption patterns emerged as data sizes varied across different percentages (30%, 50%, and 100%) of the data set employed in the proposed dynamic KG approach. When handling the full data capacity, learning demands substantially more time and energy than visualization, approximately four times as long and nearly five times the energy. Reducing data size leads to both processes becoming more efficient, but learning continues to be the more resource-intensive task. Notably, visualization maintains a consistent CPU load across all data sizes, indicating stable demand. This trend highlights the impact of data size on resource efficiency and emphasizes the importance of optimizing computational tasks, especially in energy-critical applications. The Python implementation, which covers the entire process (Fig. 2), including the chatbot tool, and the results of the energy measurement were published open source[6].

TABLE I: Energy Consumption Comparison Across Different Data Sizes: Customer Support Data set

Data size	Action	Avg. time(s) /action	Energy[Wh]	CPU Load
100%	Learning Visualization	746.60 157.85	17.36 3.64	25.93%
50%	Learning Visualization	269.55 75.24	6.29 1.72	27.03%
30%	Learning Visualization	88.83 40.81	2.06 0.94	26.16%

VII. Discussion and Outlook

The validation of knowledge graphs is a complex topic. In our case, statistical quality measures [30] are not suitable because we automatically generate KGs without a pre-defined ontology. Huaman et al. [31] proposed two approaches for the validation of KGs. In triple validation, a confidence value is calculated that uses syntactic similarity matching to evaluate whether a property value of the generated KG matches the value in an external graph. Based on this, instance validation calculates the degree to which an instance is classified as correct. This approach can only be used if there is a comparable data source. However, depending on the procedure, it is possible to validate or measure individual components. The training and testing accuracies for the proposed question and answer extraction model were 0.97 and 0.89, respectively. To improve the performance of the KG in terms of creation and search efficiency we will create subgraphs using clustering by keywords. The degree of automation is to be increased through an extraction and pre-processing component and the replacement of Rasa with an advanced chatbot. We plan to implement an active learning framework to continuously enhance the KG's accuracy and relevance. This enables the system to address areas of uncertainty or potential inaccuracies by actively seeking user feedback and additional data sources. We will also integrate explainable AI techniques to ensure transparency in the validation and expansion processes, thereby providing clear insights into the reasoning behind each modification of the KG. This dual strategy aims to refine the quality of our KG and to foster greater trust and understanding among its users.

VIII. Conclusion

We have shown how the combination of machine learning, statistics and semantic knowledge can be used to develop a support system for different domains. We applied a discriminative CRF model for information extraction and developed a feature extraction approach. With the proposed model, we recognized QA pairs to be used as input in the next phase. We used a combination of NLP and TF-IDF to build a model with corpus and similarity measure. The meaning of each term in the document is given in terms of weights representing the association between terms and problems, and a KG is created. Then, we created a continuous learning model that can update the KG with new tasks when the pattern of a question is unknown and predict an answer to problems. While simple databases could manage QA pairs, KGs excel in complex interrelationships among entities, offering dynamic updates and scalability that flat data structures lack. KGs support sophisticated semantic queries, for example in medical domain, identifying all treatments for a condition with a specific side effect, and utilize algorithms

[5]https://github.com/smfcoder/Rasa-Chatbot-with-React-JS-Interface
[6]https://gitlab.rlp.net/rgdsai/dyknograph

for enhanced data analysis capabilities. In addition, the structure of the KG allows related questions and answers to be linked so that answers to several stored questions are identified as matching a user inquiry. These functionalities are crucial for robust, complex data management and go beyond the capabilities of traditional databases.

The algorithm has been validated against data sets in the enterprise and medical domain. User feedback is used to update the knowledge graph, but also to weight possible offered answers when a similar query already exists. Through this interaction, continuous learning is enabled. The proposed approach demonstrates effective scalability and varying efficiency across different scenarios and data sizes. The data suggests that while creation is a more resource demanding process, visualization maintains consistent CPU load, indicating stable complexity.

Acknowledgment

This work was funded by the Federal Ministry for the Environment, Nature Conservation, Nuclear Safety and Consumer Protection (BMUV) projects "GreenTwin" under Grant 67KI31073A and "KIRA" under Grant 67KI32013B and supported by the Federal Ministry for Economic Affairs and Climate Action (BMWK) project "SAVE" under Grant KK5101901MS0 on the basis of a decision by the German Bundestag.

References

[1] D. Diefenbach, V. Lopez, K. Singh, and P. Maret, "Core techniques of question answering systems over knowledge bases: A survey," *Knowl. Inf. Syst.*, vol. 55, no. 3, p. 529–569, jun 2018. [Online]. Available: https://doi.org/10.1007/s10115-017-1100-y

[2] X. Bi *et al.*, "Unrestricted multi-hop reasoning network for interpretable question answering over knowledge graph," *Knowledge-Based Systems*, vol. 243, p. 108515, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0950705122002222

[3] A. Li, X. Wang, W. Wang, A. Zhang, and B. Li, "A survey of relation extraction of knowledge graphs," in *Web and Big Data*, J. Song and X. Zhu, Eds. Cham: Springer International Publishing, 2019, pp. 52–66.

[4] F. Al-Aswadi, H. Y. Chan, and K. H. Gan, "Automatic ontology construction from text: a review from shallow to deep learning trend," *Artificial Intelligence Review*, vol. 53, pp. 1–28, 08 2020.

[5] H. Cui *et al.*, "Stepwise relation prediction with dynamic reasoning network for multi-hop knowledge graph question answering," *Applied Intelligence*, vol. 53, no. 10, p. 12340–12354, sep 2022. [Online]. Available: https://doi.org/10.1007/s10489-022-04127-6

[6] H. Liu *et al.*, "Dynamic knowledge graph reasoning based on deep reinforcement learning," *Knowledge-Based Systems*, vol. 241, p. 108235, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0950705122000697

[7] J. Akroyd, S. Mosbach, A. Bhave, and M. Kraft, "Universal digital twin - a dynamic knowledge graph," *Data-Centric Engineering*, vol. 2, 2021. [Online]. Available: https://doi.org/10.1017%2Fdce.2021.10

[8] X. Huang, J. Zhang, D. Li, and P. Li, "Knowledge graph embedding based question answering," in *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, ser. WSDM '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 105–113. [Online]. Available: https://doi.org/10.1145/3289600.3290956

[9] A. Dhandapani and V. Vadivel, "Question answering system over semantic web," *IEEE Access*, vol. 9, pp. 46 900–46 910, 2021.

[10] H. Vathsala and S. G. Koolagudi, "NLP2sql using semi-supervised learning," in *Communications in Computer and Information Science*. Springer Singapore, 2021, pp. 288–299. [Online]. Available: https://doi.org/10.1007/978-981-16-0401-0_22

[11] H. Sun, T. Bedrax-Weiss, and W. W. Cohen, "Pullnet: Open domain question answering with iterative retrieval on knowledge bases and text," *CoRR*, vol. abs/1904.09537, 2019. [Online]. Available: http://arxiv.org/abs/1904.09537

[12] T. Shen *et al.*, "Multi-task learning for conversational question answering over a large-scale knowledge base," *CoRR*, vol. abs/1910.05069, 2019. [Online]. Available: http://arxiv.org/abs/1910.05069

[13] A. Abujabal, R. Saha Roy, M. Yahya, and G. Weikum, "Never-ending learning for open-domain question answering over knowledge bases," in *Proceedings of the 2018 World Wide Web Conference*, ser. WWW '18. International World Wide Web Conferences Steering Committee, 2018, p. 1053–1062. [Online]. Available: https://doi.org/10.1145/3178876.3186004

[14] L. B. Fazlic *et al.*, "A novel nlp-fuzzy system prototype for information extraction from medical guidelines," in *42nd International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, 2019, pp. 1025–1030.

[15] J. W. Wei, C. Huang, S. Vosoughi, and J. Wei, "What are people asking about covid-19? A question classification dataset," *CoRR*, vol. abs/2005.12522, 2020. [Online]. Available: https://arxiv.org/abs/2005.12522

[16] J. D. Lafferty, A. McCallum, and F. C. N. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *Proceedings of the Eighteenth International Conference on Machine Learning*, ser. ICML '01. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2001, p. 282–289.

[17] Y. Goldberg and G. Hirst, *Neural Network Methods in Natural Language Processing*. Morgan & Claypool Publishers, 2017.

[18] J. D. U. A. Rajaraman, "Data mining," in *Mining of Massive Datasets*. Cambridge University Press, 2011, pp. 1–17. [Online]. Available: https://doi.org/10.1017/cbo9781139058452.002

[19] L. Ehrlinger and W. Wöß, "Towards a definition of knowledge graphs," in *Joint Proceedings of the Posters and Demos Track of 12th International Conference on Semantic Systems - SEMANTiCS2016 and 1st International Workshop on Semantic Change & Evolving Semantics (SuCCESS16)*, vol. 1695, 09 2016.

[20] H. Paulheim, "Knowledge graph refinement: A survey of approaches and evaluation methods," *Semantic Web*, vol. 8, pp. 489–508, 2016. [Online]. Available: https://api.semanticscholar.org/CorpusID:13151033

[21] X. Chen, S. Jia, and Y. Xiang, "A review: Knowledge reasoning over knowledge graph," *Expert Systems with Applications*, vol. 141, p. 112948, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417419306669

[22] S. M. Kazemi *et al.*, "Representation learning for dynamic graphs: A survey," *J. Mach. Learn. Res.*, vol. 21, no. 1, jan 2020.

[23] F. Pedregosa *et al.*, "Scikit-learn: Machine learning in python," 2018.

[24] C. J. V. Rijsbergen, *Information Retrieval*, 2nd ed. USA: Butterworth-Heinemann, 1979.

[25] A. A. Hagberg, D. A. Schult, and P. J. Swart, "Exploring network structure, dynamics, and function using networkx," in *Proceedings of the 7th Python in Science Conference*, G. Varoquaux, T. Vaught, and J. Millman, Eds., Pasadena, CA USA, 2008, pp. 11 – 15.

[26] J. D. Hunter, "Matplotlib: A 2d graphics environment," *Computing in Science & Engineering*, vol. 9, no. 3, pp. 90–95, 2007.

[27] C. Boettiger, "cboettig/rdflib: A high level wrapper around the redland package for common rdf applications," Jan. 2018. [Online]. Available: https://doi.org/10.5281/zenodo.1134642

[28] A. Abujabal, M. Yahya, M. Riedewald, and G. Weikum, "Automated template generation for question answering over knowledge graphs," in *Proceedings of the 26th International Conference on World Wide Web*, ser. WWW '17, 2017, p. 1191–1200. [Online]. Available: https://doi.org/10.1145/3038912.3052583

[29] A. Guldner *et al.*, "Development and evaluation of a reference measurement model for assessing the resource and energy efficiency of software products and components—green software measurement model (gsmm)," *Future Generation Computer Systems*, vol. 155, pp. 402–418, 2024. [Online]. Available: https://doi.org/10.1016/j.future.2024.01.033

[30] S. Seo, H. Cheon, H. Kim, and D. Hyun, "Structural quality metrics to evaluate knowledge graphs," 2022.

[31] E. Huaman, A. Tauqeer, and A. Fensel, *Towards Knowledge Graphs Validation Through Weighted Knowledge Sources*. Springer-Verlag, 01 2021, pp. 47–60.