NL2LS: LLM-based Automatic Linking of KGs

Reda Ihtassine
reih@upb.de
Universität Paderborn

Asep Fajar Firmansyah
asep.fajar.firmansyah@upb.de
Universität Paderborn

Manzoor Ali
manzoor@upb.de
Universität Paderborn

Nikit Srivastava
nikit.srivastava@upb.de
Universität Paderborn

Mohamed Ahmed Sherif
mohamed.sherif@upb.de
Universität Paderborn

Axel-Cyrille Ngonga Ngomo
axel.ngonga@upb.de
Universität Paderborn

ABSTRACT

Integrated knowledge graphs form the foundation of numerous data-driven applications, including search engines, conversational agents, and e-commerce solutions. Declarative link discovery frameworks utilize link specifications to define the conditions necessary for establishing a link between knowledge graphs' resources. Despite domain expertise, defining such link specifications remains challenging due to their intricate syntax, threshold tuning, and the need to precisely express complex linking logic. To address this challenge, we propose NL2LS, a novel language-driven approach that leverages large language models to automatically translate natural language into link specifications, enabling domain experts and practitioners to express correct and complex linking rules more effectively. NL2LS employs three distinct training paradigms to handle the complexity of link specifications: zero-shot learning, one-shot learning and supervised fine-tuning. We evaluated NL2LS using different large language model architectures with different multi-lingual datasets. Our evaluation using metrics like BLEU, METEOR, ChrF++, and TER demonstrates that NL2LS effectively translates natural language into link specifications, lowering the technical barrier and assisting users in specifying link rules more intuitively.

CCS CONCEPTS

• **Computing methodologies** → **Semantic networks**; **Natural language generation**; **Machine translation**.

KEYWORDS

Link Specifications, Large Language Models, Natural Language Processing.

ACM Reference Format:

Reda Ihtassine, Asep Fajar Firmansyah, Manzoor Ali, Nikit Srivastava, Mohamed Ahmed Sherif, and Axel-Cyrille Ngonga Ngomo. 2018. NL2LS: LLM-based Automatic Linking of KGs. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation email (Conference acronym 'XX)*. ACM, New York, NY, USA, 8 pages. https://doi.org/XXXXXXX.XXXXXXX

Conference acronym 'XX, June 03–05, 2018, Woodstock, NY

ACM ISBN 978-1-4503-XXXX-X/2018/06…$15.00
https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

Link specifications (LSs) are a crucial component of the declarative link discovery (LD) framework, used to establish connections between entities within knowledge graphs (KGs). An LS expresses the necessary conditions to establish a link between two resources, either by using a single similarity measure (atomic LS) or use operators to aggregating multiple similarity measures (complex LS) [1]. State-of-the-art LD frameworks, such as Limes [10] and Silk [7], utilize complex LS to express the linking among knowledge graph (KG) resources. The generation of link specifications from these LD frameworks requires the involvement of LD experts, either to manually set the specifications or to employ machine learning algorithms, such as Wombat [16], to evaluate and ensure LSs are generated correctly. However, this process remains non-trivial even for experienced users and poses an even steeper challenge for non-expert users. To address this issue, *Ahmed et al.* proposed Lsvs [2, 3], a framework designed to enhance the explainability of link discovery through deep learning methods. Building on this foundation, the authors extended the model's capabilities with NMV-LS [1], which employs a large language model (LLM), such as T5. This approach aims to improve model performance by addressing the small-vocabulary issue identified in the previous work [3]. *Chen et al.* introduced LLM-Align [5], an method that Employs LLMs to semantically infer entity alignments from structured KG data.

In contrast to previous studies, our approach tackles a different but equally critical challenge: *Leveraging LLMs to assist domain experts and lay users in generating link specifications directly from English and German natural language inputs, thereby reducing the technical effort involved in formal specification.* To address this challenge, we propose an LLM-based approach for the automatic linking of KGs, called NL2LS. To assess the effectiveness of this method, we systematically evaluated LLMs under three distinct learning paradigms, zero-shot learning (ZSL), one-shot learning (OSL) and supervised fine-tuning (SFT), and included a rule-based model as a baseline for comparison. Our experimental results show that supervised fine-tuning helps LLMs learn patterns in NL and generate accurate LSs from natural language examples, with NL2LS using LOLA achieving a BLEU score of up to 98.85. Our main contributions are as follows:

- We present NL2LS, an LLM-based system capable of the automatic generation of LS from NL input.
- We evaluated our proposed approach to generate LS of two different link discovery frameworks, i.e., Limes and Silk using metrics BLEU, METEOR, ChrF++, and TER, demonstrating its effectiveness in accurately translating and aligning NL inputs with the target LS.

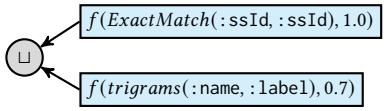

Figure 1: Complex LS with filter nodes (rectangles) applying similarity thresholds, combined by an operator node (circle) via union.

Table 1: Link Specification Syntax and Semantics [16]

LS	$[[LS]]_M$
$f(m,\theta)$	$\{(s,t)\|(s,t) \in M \wedge m(s,t) \geq \theta\}$
$L_1 \sqcap L_2$	$\{(s,t)\|(s,t) \in [[L_1]]_M \wedge (s,t) \in [[L_2]]_M\}$
$L_1 \sqcup L_2$	$\{(s,t)\|(s,t) \in [[L_1]]_M \vee (s,t) \in [[L_2]]_M\}$
$L_1 \backslash L_2$	$\{(s,t)\|(s,t) \in [[L_1]]_M \wedge (s,t) \notin [[L_2]]_M\}$

- We provided codes and data sources publicly for reproducible purposes [1].

2 PRELIMINARIES

2.1 Link Specification

A link specification (LS) formally defines conditions for generating links between KGs' resources $(s,t) \in S \times T : \rho(s,t)$ in source S and target T KGs, given a relation ρ (e.g., `owl:sameAs`). A link specification is composed of two fundamental components: (1) *Similarity Measures m*: A functions $m : S \times T \rightarrow [0,1]$ that quantify similarity between resource properties from the source set S and the target set T. Link Discovery frameworks select measures based on data type—for text, string-based measures (e.g., *Jaccard*, *Cosine*, *Trigram*) or semantic measures (e.g., *LCH*, *LI*). For example, `ExactMatch(:socId, :socId) = 1.0` if identical, while `Trigram(:name, :label) = 0.7` counts shared trigrams. Semantic measures compare meaning, such as LCH using *WordNet* taxonomies. (2) *Operators* ω: Combine similarity results into complex link specifications using intersection ($\sqcap$), union ($\sqcup$), or difference ($\backslash$) [16]. We define the semantics $[[L]]_M$ of a link specification L w.r.t. a mapping M as given in Table 1. Those semantics are defined extensionally through the mappings they generate [16]. The mapping $[[L]]$ of a link specification L with respect to $S \times T$ contains the links that will be generated by L.

A link specification can be atomic or complex: (1) *Atomic LS:* An atomic LS consists of a single filtering function, which applies a similarity measure m with a threshold θ to generate links. The filtering function can be represented as $f(m,\theta)$, where links are created only if the similarity exceeds the threshold. (2) *Complex LS:* A complex LS is formed by combining two or more atomic LSs using operators. This allows for the construction of more sophisticated conditions for linking resources. An example of a complex LS is given in Figure 1.

[1] https://github.com/dice-group/NL2LS

2.2 Encoder-Decoder Architecture for T5 Model

The T5 (Text-to-Text Transfer Transformer) [15] model (encoder-decoder) introduces a unified framework for NLP by re-framing all tasks, such as translation, summarization, classification, and question answering—into a text-to-text format. It leverages transfer learning technique by pretraining on a large-scale and fine-tuning on a diverse mixture of supervised tasks which facilitates robust performance and efficient task generalization.

2.3 Decoder Architecture for LLAMA-3 Model

LLaMA 3 [6] is a state-of-the-art foundation model developed by Meta, belonging to the class of decoder-only Generative Pre-trained Transformers (GPT). It is a family of dense transformer-based language models that integrate several advanced architectural features, including Grouped Query Attention Rotary Positional Embeddings, and SwiGLU activation functions. LLaMA 3 is trained on a massive multilingual corpus, enabling it to support extremely long context windows, exhibit strong multilingual capabilities, code generation, complex reasoning and tool usage with competitive performance in multimodal tasks (via a compositional approach).

2.4 Decoder Architecture with Integrated Mixture-of-Experts (MoE)

LOLA [17] is an open-source GPT variant designed specifically for low-resource languages. It consists of multiple decoder layers equipped with attention heads and improves both model capacity and computational efficiency by replacing the standard feed-forward networks in the Transformer architecture with MoE layers. During inference, it adopts a $top-1$ gating strategy which activates a single expert per layer, thus enabling conditional computation that balances model capacity.

3 RELATED WORK

Link specifications has been a critical component in the field of data integration, particularly in the context of the semantic web and linked data. Paganelli et al. [12] introduce LinkRule, an interactive GUI-based system that helps users create link discovery rules. By using instance data and domain ontologies to suggest property pairs and similarity functions, it makes it easier for non-experts to create high-quality link rules. LinkRule offers rule recommendations and improvements based on user input and data properties, in contrast to manual specification. Amuse [18] was introduced as an extensible and automatic instance matching framework that learns link discovery rules using machine learning and schema-based blocking. Amuse uses supervised classifiers (like decision trees) to learn rules from labeled data after generating candidate matches using heuristics. To promote openness and comprehension, the system additionally facilitates rule explanations. Recent work such as LLM-Align [5] has explored the use of LLMs to enhance entity alignment by reasoning over structured KG content, including entity names, attributes, and relations. While LLM-Align focuses on aligning individual entities across different KGs through multi-step inference and prompt engineering, it assumes the existence of candidate entities and does not explicitly generate executable alignment rules. To enhance link discovery across heterogeneous data sources, Volz et al. [8] suggest a hybrid strategy that combines

statistical instance matching techniques with link keys, which are schema-level rules for identifying equivalent entities. Nikolov et al. [11] proposed an active learning-based approach for link discovery, which automatically generates link specifications by iteratively refining them based on user feedback. In contrast to prior works which rely on graphical interfaces, labeled data, or ontology-driven strategies, our approach introduces a novel paradigm by enabling the direct generation of link specifications from natural language using LLMs.

The increasing complexity of data on the web has led to the development of hybrid approaches that combine rule-based and learning-based methods. One notable example is the work by Ngomo and Auer [9], who introduced the Limes framework for link discovery [10]. Limes allows users to define link specifications using a declarative language, and it optimizes the link discovery process by utilizing mathematical properties of metric spaces. On the other hand, the Silk framework [20] offers a flexible and comprehensive approach to link specifications. Silk allows users to create detailed LSs through a declarative language, enabling the definition of complex linkage rules between datasets. Recent research on multilingual verbalization systems, such as [1–3], has demonstrated significant progress in translating link specification into natural language using sophisticated encoder-decoder architectures, including GRU, LSTM, and transformer models as well as LLMs. These advancements are further enhanced by the integration of few-shot learning techniques and the T5 model, leading to improved performance in generating verbalizations across different languages. In contrast, the work we introduce in this paper addresses the inverse problem (i.e, generating LSs from NL).

4 APPROACH

To effectively translate natural language into link specifications, we explored a multi-faceted approach utilizing LLMs. Our methodology encompassed two distinct settings paradigms which are rule-based and LLM-based approach. In particular, the LLM-based approach considered the techniques of zero-shot learning, one-shot learning and supervised fine-tuning. A detailed description of the datasets employed in our experiments is provided in section 5.1.

4.1 Rule-based (Baseline)

We begin by considering the most straightforward solution of the problem, we implemented a rule-based parser that translates English and German natural language descriptions into formal link specifications using handcrafted Python regular expressions (*regex*). The parser identifies patterns in the input text, such as similarity functions (e.g., *Levenshtein*, *Qgrams*, *Overlap*) and their associated thresholds and maps them to a LS syntax. This approach allows us to systematically extract structured LS expressions from semi-structured natural language inputs, facilitating automated evaluation and model benchmarking. This model serves as a baseline to assess the effectiveness of deterministic approaches and to evaluate whether such methods are sufficient for the NL-to-LS translation task.

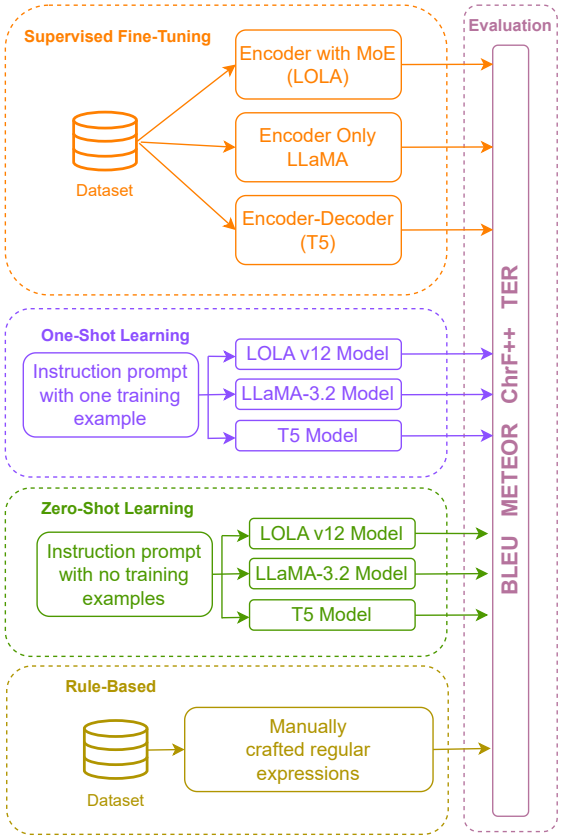

Figure 2: NL2LS architecture.

4.2 LLM-based Approach

We investigated distinct architectures for the LLMs to explore their effectiveness in translating natural language into link specifications. The LLM architectures considered have been discussed in sections 2.2 to 2.4.

Zero-Shot-Learning (ZSL). In the initial phase, we implement ZSL approach, where the model is prompted to translate NL to LS with absence of any task-specific examples during inference. In contrast to supervised or few-shot methods, ZSL uses the generalization skills that the LLM acquired during pretraining to understand and execute the task based only on instructions given in the prompt. The prompt consists solely of an instruction that describes the task (e.g., *"Translate the following natural language sentence into a link specification based on the LIMES Link Discovery Framework": ADD(0.5*MAX(Jaro(a.rdfs:label, b.rdfs:label)), 0.5*WGS84(a.wgs84:lat, b.wgs84:lat))*). This prompt is then provided to the pre-trained, non-fine-tuned model, which generates the corresponding link specifications based purely on its prior knowledge.

```
Prompt question = Next is a natural question and its translation
      to link specification:

Natural Language Specification: a link will be constructed
      whereverthe firstNames of the source and the target have a
      Qgrams similarity of 62% and the authors of the source and
      the target have a Jaccard similarity of 25% and the
      firstNames of the source and the target have a Qgrams
      similarity of 25% and the authors of the source and the
      target have a Ratcliff similarity of 62% and the firstNames
      of the source and the target have a Mongeelkan similarity of
      62% and the authors of the source and the target have a
      Qgrams similarity of 25%
Corresponding Link Specification: AND(AND(AND(qgrams(x.firstName,y
      .firstName)|0.62,AND(AND(jaccard(x.authors,y.authors)|0.25,
      qgrams(x.firstName,y.firstName)|0.25)|0.62,ratcliff(x.authors
      ,y.authors)|0.62)|0.25)|0.25,mongeElkan(x.firstName,y.
      firstName)|0.62)|0.62,qgrams(x.authors,y.authors)|0.25)

Train on it and translate next natural language questions to their
       link spec (give each response in a line):

Natural Language Specification: a link will be produced when the
      name of the source and the movie of the target have a Cosine
      similarity of 45% or a Qgrams similarity of 25% or the movies
       of the source and the target have an Overlap similarity of
      45% or the name of the source and the movie of the target
      have a Soundex similarity of 25% or the movies of the source
      and the target have a Qgrams similarity of 25%
```

Listing 1: One-shot learning prompt example

One-Shot-Learning (OSL). Following the initial phase we implement OSL approach to provide the models with a minimalistic training setup. In OSL the model is trained to recognize new classes or categories with only a single training example, or very few examples, per class. This contrasts with traditional machine learning models, which typically require large amounts of labeled data to perform well [19]. The generated prompt for OSL include a natural language sentence and its corresponding link specification (Listing 1 provides an example prompt). After being exposed to just this one data sample, the LLM is then asked to generate the corresponding link specification for the test NL question. This approach aims to enhance the models' ability to extrapolate and apply the learned pattern from the lone training instance to new unseen prompts.

Supervised Fine-Tuning (SFT). The final stage of our training regime is supervised fine-tuning. In this phase, as described in Figure 2, we perform supervised fine-tuning of the T5-small, LLaMA-3.2B, and LOLA.v2 models using the full training and validation datasets, leveraging labeled input–output pairs. This process involves systematically adjusting the models' parameters to optimize their performance specifically for our task. By exposing the models to the full spectrum of training data, we aim to achieve a deeper understanding of the translation patterns, thereby significantly improving the models' accuracy and reliability in producing correct link specifications.

5 EVALUATION

To evaluate the performance across these different training approaches, models, and datasets, we employed a suite of testing metrics. These include BLEU, METEOR, ChrF++, and TER. Through this evaluation, we aim to identify the strengths and weaknesses of each model and training paradigm.

Table 2: Statistics of datasets used in our experiments. Atomic LS (aLS) categories: aLS ≤ 4, $4 <$ aLS ≤ 6, and aLS > 6.

Dataset	Total	aLS ≤ 4	$4 <$ aLS ≤ 6	aLS > 6
LIMES-S	15,000	2,162	12,838	–
LIMES-SM	137	19	86	32
SILK-S	146	126	7	13
LIMES-A	100	13	87	–
LIMES-GT	60	–	60	–
LIMES-M	8	8	–	–
SILK-A	8	8	–	–

5.1 Datasets

In our experiments, we leverage the benchmark datasets introduced in [1], including `Limes-Silver`, `Limes-Human-Annotated`, `Limes-Human-Annotated`-manipulated (hereafter abbreviated as `Limes-manipulated`) and `Silk`-human-annotated. However, due to the limited size of the latter two datasets, we extend both the `Limes`-manipulated and `Silk-human-annotated` datasets, and we produced `Silk-Silver`. Additionally, we include the `Limes`-geo-temporal benchmark, a more complex dataset comprising link specifications that capture geo-spatial proximity, spatial neighborhood relations, and temporal constraints. Furthermore, we extended the benchmark by creating a German counterpart, in which the natural language descriptions were translated into German while preserving the original link specifications.

As described in 2.1, link specification can be atomic or complex. Table 2 provides statistical information about the datasets and illustrates atomic link specifications distribution for each dataset.

The details of the datasets are as follows:

- `Limes-Silver`: This dataset comprises silver-standard LIMES datasets, consisting of link specifications and their verbalizations automatically generated using a rule-based verbalizer [3].
- `Limes-Annotated`: Derived from Limes-Silver, this dataset has been curated by human annotators to ensure the accuracy and correctness of the verbalizations associated with each link specification. It contains 100 LSs, providing a high-quality, human-verified subset for precise evaluation. The careful human curation ensures that the LIMES-Annotated dataset serves as a reliable benchmark in comparison to automatically generated datasets.
- `Limes-Manipulated`: This is an extended version of Limes-Silver, in which human annotators modify the patterns of link specifications and provide their verbalizations. The dataset is relatively small but offers a human intervention in refining link specification patterns, offering a valuable Evaluation resource.
- `Silk-Annotated`: This dataset is based on the SILK link specification framework [7] Human annotators have been involved in carefully providing accurate verbalizations for the link specifications, ensuring high-quality, human-verified outputs. Despite its small size, the SILK-Annotated dataset serves as a focused and reliable evaluation resource within the SILK framework.

- `Limes-Geo-Temporal`: This dataset contains LS that express both distance-based conditions over geographic properties, used to determine whether two spatial entities in a KG are sufficiently close to be considered a match, and temporal conditions that link entities based on time-related properties, such as events occurring simultaneously or with overlapping durations.

5.2 Evaluation Metrics

In our work, we evaluate our approach using the following automatic evaluation metrics: (1) The *BLEU* (Bilingual evaluation understudy) [13] is used to evaluate the quality of text translations between natural languages with higher scores indicating better performance. (2) *METEOR* (Metric for Evaluation of Translation with Explicit ORdering) [4] evaluates machine translation quality by calculating the harmonic mean of uni-gram precision and recall, with a higher weight assigned to recall than to precision, where higher scores indicate better performance. (3) *ChrF++* [14] stands for Character *n*-gram F-score, which measures the harmonic mean of precision and recall, where Higher scores are better. (4) *TER* (Translation Edit Rate) measures the number of edits required to change a system output into one of the references, where lower scores indicate better performance.

5.3 Experimental Setup

To evaluate our model for generating link specifications from textual input, we performed several experiments using different benchmark patterns to address the following research questions:

- Q_1. To what extent could a rule-based (baseline) system solve the problem of translating NL into LS?
- Q_2. How effective are LLMs in their default zero-shot learning setting at transforming natural language inputs into link specifications?
- Q_3. Does one-shot learning enhance the performance of LLMs in converting natural language to link specification?
- Q_4. Does fine-tuning the LLMs improve its performance in translating natural language to link specification?
- Q_5. Does the best fine-tuned system generalize to solve all types of the task of translating natural language to link specification?

Dataset Splits. For the supervised fine-tuning experiments, we used multiple benchmark datasets introduced in the evaluation (see Section 5) with varying levels of link specification complexity. We adopted a standard 80/10/10 split for training, validation, and testing, respectively. All data were processed into a tab-separated format with natural language descriptions as inputs and their corresponding link specifications as targets.

Fine-tuning T5 (Encoder-Decoder). To fine-tune the T5 model for the NL-to-LS translation task, we used the t5-small encoder-decoder architecture with a standard text-to-text formulation. Each input consisted of a natural language sentence prefixed with the prompt *"Translate text into Link Specification:"*, while the target output was the corresponding link specification. Inputs were tokenized with a maximum length of 512 tokens and targets with a maximum length of 256 tokens. Padding tokens in the labels were replaced with −100 to properly ignore them during loss computation. We used tab-separated files for training, validation, and testing (`train.txt`, `dev.txt`, `test.txt`) with two columns: `source` (NL) and `target` (LS). These were converted into Hugging Face Dataset objects. The model was fine-tuned using the Hugging Face Seq2SeqTrainer with the following hyperparameters: 5 training epochs, a learning rate of 2e-4, batch size of 4 for both training and evaluation, weight decay of 0.01, and a save strategy that retains up to 2 checkpoints per epoch. Mixed-precision training was disabled. The training process used DataCollatorForSeq2Seq for dynamic padding and the predict_with_generate flag enabled sequence generation during evaluation.

Fine-tuning LLaMA/LOLA (Decoder-only). For the decoder-only architectures we followed two distinct setups. LLaMA model was fine-tuned using Hugging Face Transformers with 4-bit quantization (nf4) via BitsAndBytes to reduce memory consumption. The training utilized the AutoModelForCausalLM class and causal language modeling objectives, where each input was formatted as a chat-style prompt. Inference was conducted using the Hugging Face pipeline interface. For LOLA, we applyed Parameter-Efficient Fine-Tuning (PEFT) using the LoRA method. Specifically, we enabled low-rank adapters for target modules with LoRA rank and alpha set to 32 and dropout to 0.1. The model was trained using the Hugging Face Trainer class where sources and targets were combined and tokenized to produce masked labels for supervised learning. During inference, pre-trained LoRA adapters were loaded, and predictions were generated in batch from TSV files using a prompt template. All models were trained with a maximum sequence length of 512 tokens.

5.4 Results & Analysis

To answer Q_1 we begin by exploring the most straightforward baseline: a rule-based model. The results of this approach are presented in Table 3. While our rule-based approach provides a transparent and interpretable baseline, its performance is inherently limited by its inability to generalize beyond predefined rules, especially when faced with complex or ambiguous natural language inputs. In particular, the rule-based model performed poorly on both English and German datasets achieving its best performance on the Limes-Geo-Temporal (German) dataset with a BLEU score of 20.3. This limited performance can be attributed to the rigid nature of handcrafted regular expressions, which struggle to handle the syntactic and semantic variability of natural language, especially in complex and diverse sentence structures. This insight addresses our first research question Q_1 and underscores the need for more advanced methods. Consequently, we developed a subsequent solution based on large language models (LLMs).

To address research question Q_2, we developed an LLM-based solution leveraging zero-shot learning. The results of this approach are summarized in Table 4. Notably, all evaluated LLMs consistently under-perform compared to the rule-based baseline across all datasets in zero-shot settings. In particular, the T5 model performed poorly across all datasets, achieving a BLEU score of 0.0 on all of the German datasets and a modest score of 4.2 on Limes-silver dataset.

These results indicate that T5 struggles to generalize effectively for this task under zero-shot conditions. While the LLaMA model exhibited a slight improvement over T5, its overall performance remained notably limited—achieving a maximum BLEU score of 5.4 on the English Limes-Silver dataset and 5.3 on its German counterpart. However, LLaMA failed to generate any valid link specifications on the remaining datasets, with BLEU scores of 0.0 across the board, highlighting its limited capability for this task. The LOLA model showed a marginal improvement over LLaMA, achieving a BLEU score of 11.4 on the Limes-Silver dataset. However, it also failed to generate valid link specifications on the remaining datasets, indicating limited capability under zero-shot learning scenarios. These results highlight that, despite the strengths of encoder-decoder, decoder-only, and decoder architectures with Mixture-of-Experts (MoE) layers, these models remain insufficiently effective for the specific task of translating natural language into link specifications in zero-shot settings.

Table 3: Performance of the rule-based Model for the NL to LS translation task on the English (EN) and German (DE) dataset. Dataset labels: GT = Geo-Temporal, S = Silver, HA = Human Annotated, LM = Limes Manipulated, SM= Silver Manipulated.

	EN				DE			
Dataset	BLEU	METEOR	ChrF++	TER	BLEU	METEOR	ChrF++	TER
LIMES-GT	1.0	0.0	0.1	0.9	**20.3**	0.2	0.3	0.6
LIMES-S	13.4	0.1	0.2	0.7	0.0	0.0	0.1	1.4
LIMES-HA	13.1	0.1	0.2	0.7	0.0	0.0	0.1	1.2
LIMES-M	**15.8**	0.2	0.2	0.8	0.0	0.0	0.2	1.7
SILK-HA	0.0	0.1	0.3	1.0	0.0	0.0	0.1	1.4
LIMES-SM	7.2	0.1	0.2	1.0	0.0	0.1	0.1	1.4
SILK-S	0.0	0.1	0.1	1.3	0.0	0.0	0.1	1.4

To address research question Q_3, we present the results from the one-shot learning of LLMs experiment, shown in Table 5. The performance of the LLaMA-3 model varies significantly across different datasets. It performs best on the Limes-Manipulated dataset, achieving the highest BLEU score of 91.8. However, LLaMA-3 struggles with the Limes-Geo-Temporal dataset, indicating difficulties in producing accurate translations for this specific dataset. The T5 model reveals a moderate and consistent performance across all datasets with the highest BLEU score of 79.8 on the Silk-Human-Annotated dataset. The LOLA model outperforms T5 and LLaMA-3's performance across all datasets with the highest BLEU score of 93.7 for the Silk-Human-Annotated. These results demonstrate that LLMs can perform much better when given minimal learning examples, specifically through one-shot learning. Nevertheless, even with high accuracy scores, we are unable to draw the conclusion that the generated link specifications are effective enough. Instead of indicating complete correctness, these scores mainly show superficial similarity to the reference specifications. To guarantee their validity in practice, link specifications must exhibit a higher level of accuracy. Therefore, human expertise is still necessary to reliably verify and implement the generated link specifications.

Table 4: Performance of LLMs in Zero-Shot-Learning scenario for the NL to LS task in English (EN) and German (DE), using T5, LLaMA-3 and LOLA. Dataset labels: GT = Geo-Temporal, S = Silver, HA = Human Annotated, LM = Limes Manipulated, SM = Silver Manipulated.

		EN				DE			
Model	**Dataset**	BLEU	METEOR	ChrF++	TER	BLEU	METEOR	ChrF++	TER
T5	LIMES-GT	0.0	0.1	0.2	0.0	0.0	0.1	0.1	1.0
	LIMES-S	**4.2**	0.1	0.2	1.7	0.0	0.0	0.1	1.1
	LIMES-HA	0.0	0.0	0.2	1.4	0.0	0.0	0.1	1.0
	LIMES-M	0.0	0.1	0.3	1.6	0.0	0.0	0.1	1.4
	SILK-HA	0.0	0.0	0.3	1.0	0.0	0.0	0.1	1.0
	LIMES-SM	0.0	0.0	0.1	1.0	0.0	0.0	0.1	1.0
	SILK-S	0.0	0.0	0.1	1.3	0.0	0.0	0.1	1.1
LLaMA	LIMES-GT	0.0	0.0	0.1	1.1	0.0	0.0	0.1	1.0
	LIMES-S	**5.4**	0.1	0.3	1.1	**5.3**	0.1	0.3	0.9
	LIMES-HA	0.0	0.1	0.3	0.9	0.0	0.0	0.0	1.2
	LIMES-M	0.0	0.1	0.2	1.3	0.0	0.0	0.1	1.1
	SILK-HA	0.0	0.0	0.1	0.9	0.0	0.0	0.1	1.0
	LIMES-SM	0.0	0.1	0.2	1.1	0.0	0.0	0.1	1.2
	SILK-S	0.0	0.0	0.1	1.1	0.0	0.0	0.1	2.1
LOLA	LIMES-GT	0.0	0.0	0.1	1.0	1.5	0.0	0.2	1.2
	LIMES-S	**11.4**	0.1	0.2	1.5	**1.8**	0.1	0.1	1.0
	LIMES-HA	0.0	0.0	0.1	1.1	0.0	0.0	0.1	1.0
	LIMES-M	0.0	0.0	0.3	1.8	0.0	0.0	0.1	1.2
	SILK-HA	0.0	0.0	0.1	1.1	0.0	0.0	0.1	1.2
	LIMES-SM	0.0	0.0	0.1	1.0	0.0	0.0	0.1	1.0
	SILK-S	0.0	0.0	0.0	1.1	0.0	0.0	0.1	1.2

To address research question Q_4, we fine-tuned the LLMs used in the previous experiment on the benchmark datasets described earlier. The results are reported in Table 6. The performance of the LOLA and LLaMA models stands out, yielding promising results and achieving the highest scores across the majority of benchmark datasets. Notably, both models excel on the English versions of the Limes-Silver dataset, achieving a BLEU score of 98.8. On the German version of the Limes-Silver dataset, LLaMA delivers strong results with a BLEU score of 97.7, while LOLA also performs competitively, achieving a BLEU score of 93.0. Unlike LOLA, LLaMA additionally demonstrates high performance on the English Limes-Human-Annotated dataset, with a BLEU score of 97.3. However, their performance on the most German datasets is noticeably lower than on the English counterparts, with BLEU scores ranging from 11.4 for the Limes-Geo-Temporal dataset to 77.5 for the Limes-Annotated dataset. In contrast, the T5 model demonstrates moderate performance across all datasets, with the exception of the Limes-Silver dataset, where it achieved a relatively high BLEU score of 79.9. T5 achieves a BLEU score of 37.2 on the Limes-Annotated dataset and 26.0 on Silk-Silver, demonstrating also a significant lower scores compared to LLaMA and LOLA. The modest performance of the T5 model can be attributed to its encoder-decoder

Table 5: Performance of LLMs in One-Shot-Learning scenario for the NL to LS task in English (EN) and German (DE), using T5, LLaMA-3 and LOLA. Dataset labels: GT = Geo-Temporal, S = Silver, HA = Human Annotated, LM = Limes Manipulated, SM = Silver Manipulated.

		EN				DE			
Model	**Dataset**	BLEU	METEOR	ChrF++	TER	BLEU	METEOR	ChrF++	TER
T5	LIMES-GT	32.2	0.3	0.5	0.5	24.5	0.2	0.4	0.6
	LIMES-S	58.7	0.4	0.5	0.5	50.0	0.3	0.4	0.5
	LIMES-HA	53.2	0.3	0.4	0.5	45.0	0.3	0.4	0.6
	LIMES-M	60.7	0.4	0.7	0.8	55.0	0.4	0.5	0.5
	SILK-HA	**79.8**	0.6	0.7	0.5	**74.0**	0.5	0.6	0.4
	LIMES-SM	29.0	0.3	0.4	0.6	31.0	0.3	0.4	0.6
	SILK-S	33.0	0.3	0.5	0.5	40.0	0.3	0.5	0.5
LLaMA	LIMES-GT	21.3	0.2	0.4	0.6	18.8	0.2	0.3	0.7
	LIMES-S	73.9	0.4	0.6	0.3	72.2	0.6	0.7	0.3
	LIMES-HA	70.1	0.4	0.6	0.4	70.5	0.5	0.6	0.3
	LIMES-M	**91.8**	0.7	0.9	0.4	**91.2**	0.7	0.7	0.2
	SILK-HA	57.4	0.5	0.6	0.6	54.8	0.4	0.5	0.4
	LIMES-SM	49.0	0.4	0.5	0.4	46.0	0.4	0.5	0.4
	SILK-S	43.0	0.4	0.5	0.5	40.0	0.3	0.5	0.5
LOLA	LIMES-GT	17.1	0.2	0.3	0.7	23.8	0.2	0.4	0.6
	LIMES-S	51.6	0.4	0.5	0.5	50.3	0.4	0.5	0.5
	LIMES-HA	76.2	0.6	0.7	0.3	77.6	0.6	0.7	0.3
	LIMES-M	71.9	0.6	0.6	0.3	72.5	0.6	0.7	0.3
	SILK-HA	**93.7**	0.7	0.8	0.2	**91.7**	0.7	0.8	0.2
	LIMES-SM	65.6	0.5	0.6	0.4	64.1	0.5	0.6	0.4
	SILK-S	54.6	0.4	0.5	0.4	56.4	0.4	0.6	0.4

Table 6: Evaluating the supervised fine-tuning LLMs for the NL to LS task in English (EN) and German (DE), using T5, LLaMA-3 and LOLA. Dataset labels: GT = Geo-Temporal, S = Silver, HA = Human Annotated, LM = LIMES Manipulated, SM = Silver Manipulated.

		EN				DE			
Model	**Dataset**	BLEU	METEOR	ChrF++	TER	BLEU	METEOR	ChrF++	TER
T5	LIMES-GT	17.5	0.2	0.2	0.7	16.4	0.2	0.2	0.7
	LIMES-S	**79.9**	0.6	0.7	0.5	**62.3**	0.4	0.8	0.7
	LIMES-HA	50.4	0.3	0.5	0.5	37.2	0.2	0.3	1.0
	LIMES-M	18.4	0.1	0.2	1.2	17.9	0.1	0.2	1.2
	SILK-HA	18.1	0.1	0.2	1.2	17.4	0.1	0.2	1.2
	LIMES-SM	16.8	0.1	0.2	1.3	15.9	0.1	0.2	1.3
	SILK-S	26.0	0.2	0.2	1.1	23.6	0.2	0.2	1.1
LLaMA	LIMES-GT	27.5	0.3	0.3	0.5	34.9	0.3	0.4	0.6
	LIMES-S	**98.8**	0.7	0.8	0.0	**97.7**	0.7	0.9	0.0
	LIMES-HA	97.3	0.7	0.9	0.1	55.9	0.4	0.5	0.4
	LIMES-M	63.7	0.5	0.6	0.6	32.6	0.3	0.3	0.7
	SILK-HA	31.2	0.2	0.3	0.7	24.1	0.2	0.3	1.0
	LIMES-SM	15.1	0.2	0.2	1.4	16.4	0.2	0.2	1.2
	SILK-S	48.2	0.3	0.5	0.5	28.7	0.2	0.3	1.0
LOLA	LIMES-GT	18.8	0.2	0.2	0.6	11.4	0.1	0.2	1.5
	LIMES-S	**98.8**	0.7	0.8	0.0	**93.0**	0.6	0.8	0.1
	LIMES-HA	74.5	0.5	0.6	0.2	77.5	0.6	0.7	0.4
	LIMES-M	23.6	0.7	0.2	1.1	22.5	0.2	0.2	1.0
	SILK-HA	21.8	0.2	0.2	1.1	19.9	0.2	0.2	1.2
	LIMES-SM	24.8	0.2	0.3	1.0	18.9	0.1	0.2	1.2
	SILK-S	22.0	0.1	0.2	1.1	21.0	0.1	0.2	1.2

architecture, which, while effective for text-to-text tasks, may introduce additional overhead or limitations in generating link specifications compared to decoder-only and decoder with MoE architectures. These results clearly demonstrate that supervised fine-tuning significantly enhances model performance compared to rule-based baseline, zero-shot learning and one-shot learning approaches, especially for datasets in English language with moderate complexity. This analysis answers our fourth research question by highlighting that while supervised fine-tuned LLMs perform well on moderately complex natural language inputs, its effectiveness diminishes as the complexity increases, underscoring the ongoing challenges in handling intricate natural language inputs.

To further investigate the robustness and cross-dataset generalization of the fine-tuned LLM, we conducted a set of transfer evaluations. To address research question Q_5, we assessed the generalization capability of our fine-tuned LOLA model that was trained on the Limes-Silver dataset, by evaluating its performance on the remaining benchmark datasets. The corresponding results are presented in Table 7. Specifically, we applied natural language inputs from the other datasets directly to the fine-tuned LOLA model, prompting it to generate the corresponding link specifications. Remarkably, the model demonstrated strong generalization, achieving a BLEU score of 98.7 on the English Limes-Annotated dataset and 95.5 on the German Silk-Silver dataset. This analysis answers our fifth research question by demonstrating the model's strong generalizability in translating diverse types of natural language into accurate link specifications.

Table 7: Performance of fine-tuned LOLA model (trained on Limes-Silver dataset) on English (EN) and German (DE) benchmark datasets.

	EN				DE			
Dataset	BLEU	METEOR	ChrF++	TER	BLEU	METEOR	ChrF++	TER
LIMES-GT	91.3	0.7	0.7	0.0	92.1	0.7	0.8	0.0
LIMES-HA	**98.7**	0.7	0.8	0.0	91.7	0.7	0.8	0.0
LIMES-M	93.4	0.7	0.8	0.0	93.8	0.7	0.8	0.0
SILK-HA	94.1	0.7	0.8	0.0	94.4	0.7	0.8	0.0
LIMES-SM	94.7	0.7	0.8	0.0	95.0	0.7	0.8	0.0
SILK-S	95.2	0.7	0.8	0.0	**95.5**	0.7	0.8	0.0

6 CONCLUSION & FUTURE WORK

This paper presents NL2LS, an LLM-based approach for the automatic generation of link specifications from natural language inputs in both English and German. To this end, we investigate two distinct settings paradigms: rule-based and LLM-based approach. Within the LLM-based approach, we investigate zero-shot learning, one-shot learning and supervised fine-tuning. In the zero-shot

setting, we exploit the inherent capabilities of pre-trained LLMs to generate link specifications without task-specific training. While this approach demonstrates limited success for simple natural language inputs, empirical evaluation reveals a substantial decline in performance as the syntactic or semantic complexity of the inputs increases. In contrast, one-shot learning introduces a minimal example to guide the LLMs, resulting in noticeably higher performance relative to zero-shot learning. The supervised fine-tuning paradigm involves training LLMs on datasets comprising natural inputs and their corresponding LSs in both English and German. This method yields significant performance gains, particularly for models such as LLaMA and LOLA, which consistently outperform T5 across multiple benchmark datasets. In conclusion, the proposed system assists experts in producing accurate link specifications that remain easily modifiable, thereby enhancing their practical usability.

In further work we address the limitations observed with complex natural language inputs and datasets. Also, we plan to enhance the robustness and generalization of NL2LS. We intend to perform Retrieval-Augmented Generation (RAG) to further improve the accuracy and contextual relevance of LS generation. In this setup, our fine-tuned LLMs will retrieve relevant examples or schema-level information from external sources such as the LIMES and SILK frameworks to ensure higher data reliability and better learning outcomes. Additionally, we plan to extend the benchmark datasets to include additional languages beyond English and German. By addressing these challenges, we aim to further improve the NL2LS system and unlock its full potential for automating complex LS generation tasks.

REFERENCES

[1] Abdullah Fathi Ahmed, Asep Fajar Firmansyah, Mohamed Ahmed Sherif, Diego Moussallem, and Axel-Cyrille Ngonga Ngomo. 2023. Explainable Integration of Knowledge Graphs Using Large Language Models. In *Natural Language Processing and Information Systems*, Elisabeth Métais, Farid Meziane, Vijayan Sugumaran, Warren Manning, and Stephan Reiff-Marganiec (Eds.). Springer Nature Switzerland, Cham, 124–139.

[2] Abdullah Fathi Ahmed, Mohamed Ahmed Sherif, Diego Moussallem, and Axel-Cyrille Ngonga Ngomo. 2021. Multilingual Verbalization and Summarization for Explainable Link Discovery. *Data and Knowledge Engineering* 133 (2021), 101874. https://doi.org/10.1016/j.datak.2021.101874

[3] Abdullah Fathi Ahmed, Mohamed Ahmed Sherif, and Axel-Cyrille Ngonga Ngomo. 2019. LSVS: Link Specification Verbalization and Summarization. In *Natural Language Processing and Information Systems*, Elisabeth Métais, Farid Meziane, Sunil Vadera, Vijayan Sugumaran, and Mohamad Saraee (Eds.). Springer International Publishing, Cham, 66–78.

[4] Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for MT and/or Summarization* (Ann Arbor, Michigan). ACL, 65–72.

[5] Xuan Chen, Tong Lu, and Zhichun Wang. 2024. LLM-Align: Utilizing Large Language Models for Entity Alignment in Knowledge Graphs. *arXiv preprint arXiv:2412.04690* (2024).

[6] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783* (2024).

[7] Robert Isele, Anja Jentzsch, and Christian Bizer. 2010. Silk Server - Adding missing Links while consuming Linked Data. In *1st International Workshop on Consuming Linked Data*.

[8] Chloé Khadija Jradeh, Ensiyeh Raoufi, Jérôme David, Pierre Larmande, François Scharffe, Konstantin Todorov, and Cassia Trojahn. 2025. Graph Embeddings Meet Link Keys Discovery for Entity Matching. In *Proceedings of the ACM on Web Conference 2025*. 3344–3353.

[9] Axel-Cyrille Ngonga Ngomo. 2011. A Time-Efficient Hybrid Approach to Link Discovery. In *Proceedings of OM@ISWC*.

[10] Axel-Cyrille Ngonga Ngomo, Mohamed Ahmed Sherif, Kleanthi Georgala, Mofeed Hassan, Kevin Dreßler, Klaus Lyko, Daniel Obraczka, and Tommaso Soru. 2021. LIMES - A Framework for Link Discovery on the Semantic Web. *KI-Künstliche Intelligenz, German Journal of Artificial Intelligence - Organ des Fachbereichs "Künstliche Intelligenz" der Gesellschaft für Informatik e.V.* (2021). https://papers.dice-research.org/2021/KI_LIMES/public.pdf

[11] Andriy Nikolov, Mathieu d'Aquin, and Enrico Motta. 2012. Unsupervised learning of link discovery configuration. In *The Semantic Web: Research and Applications: 9th Extended Semantic Web Conference, ESWC 2012, Heraklion, Crete, Greece, May 27-31, 2012. Proceedings 9*. Springer, 119–133.

[12] Matteo Paganelli, Francesco Del Buono, Andrea Baraldi, Francesco Guerra, et al. 2022. Analyzing how BERT performs entity matching. *Proceedings of the VLDB Endowment* 15, 8 (2022), 1726–1738.

[13] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on Association for Computational Linguistics*.

[14] Maja Popović. 2017. chrF++: words helping character n-grams. In *Proceedings of the Second Conference on Machine Translation*. 612–618.

[15] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *Journal of Machine Learning Research* 21, 140 (2020), 1–67.

[16] Mohamed Ahmed Sherif, Axel-Cyrille Ngonga Ngomo, and Jens Lehmann. 2017. WOMBAT - A Generalization Approach for Automatic Link Discovery. Springer.

[17] Nikit Srivastava, Denis Kuchelev, Tatiana Moteu Ngoli, Kshitij Shetty, Michael Roeder, Hamada Zahera, Diego Moussallem, and Axel-Cyrille Ngonga Ngomo. 2025. LOLA – An Open-Source Massively Multilingual Large Language Model. In *Proceedings of the 31st International Conference on Computational Linguistics*, Owen Rambow, Leo Wanner, Marianna Apidianaki, Hend Al-Khalifa, Barbara Di Eugenio, and Steven Schockaert (Eds.). Association for Computational Linguistics, Abu Dhabi, UAE, 6420–6446. https://aclanthology.org/2025.coling-main.428/

[18] Danai Symeonidou, Vincent Armant, and Nathalie Pernelle. 2020. BECKEY: understanding, comparing and discovering keys of different semantics in knowledge bases. *Knowledge-Based Systems* 195 (2020), 105708.

[19] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. 2016. Matching Networks for One Shot Learning. In *Advances in neural information processing systems*.

[20] Julius Volz, Christian Bizer, Martin Gaedke, and Georgi Kobilarov. 2009. Silk-a link discovery framework for the web of data. *Ldow* 538 (2009), 53.