mRAKL: Multilingual Retrieval-Augmented Knowledge Graph Construction for Low-Resourced Languages.

Hellina Hailu Nigatu
UC Berkeley
hellina_nigatu@berkeley.edu

Min Li
Apple
min_li6@apple.com

Maartje ter Hoeve
Apple
m_terhoeve@apple.com

Saloni Potdar
Apple
s_potdar@apple.com

Sarah E. Chasins
UC Berkeley
schasins@berkeley.edu

Abstract

Knowledge Graphs represent real-world entities and the relationships between them. Multilingual Knowledge Graph Construction (mKGC) refers to the task of automatically constructing or predicting missing entities and links for knowledge graphs in a multilingual setting. In this work, we reformulate the mKGC task as a Question Answering (QA) task and introduce mRAKL: a Retrieval-Augmented Generation (RAG) based system to perform mKGC. We achieve this by using the head entity and linking relation in a question, and having our model predict the tail entity as an answer. Our experiments focus primarily on two low-resourced languages: Tigrinya and Amharic. We experiment with using higher-resourced languages Arabic and English for cross-lingual transfer. With a BM25 retriever, we find that the RAG-based approach improves performance over a no-context setting. Further, our ablation studies show that with an idealized retrieval system, mRAKL improves accuracy by 4.92 and 8.79 percentage points for Tigrinya and Amharic, respectively.

1 Introduction

Knowledge Graphs (KG) are structured multi-relational graphs that store factual knowledge. In a KG, nodes represent entities (e.g., Michelle Obama, Sasha Obama) and links represent relationships between the nodes (e.g., Michelle Obama - mother - Sasha Obama). Multilingual KGs are KGs in multiple languages.

Despite their myriad downstream applications, including Question Answering (Huang et al., 2019; Jiang et al., 2023), Information Retrieval (Reinanda et al., 2020), and Language Model Augmentation (Tian et al., 2024; Wu et al., 2022), most KGs are incomplete (Saxena et al., 2022a; Zhou et al., 2022). The quantity of missing information in KGs is even greater in low-resourced languages (Zhou et al., 2022). Additionally, manual construction of KGs is expensive (Paulheim, 2018). Recent work has investigated the use of pre-trained Language Models (LMs) for KG Construction (e.g. Saxena et al., 2022a; Yao et al., 2019). However, most of the work is focused on English, for which LMs have good performance (Zhou et al., 2022).

Multilingual Knowledge Graph Construction (mKGC) research allows us to (1) extend the downstream benefits of KGs to multiple languages, and (2) capture culturally nuanced and relevant information across languages. However, the challenges of mKGC are exacerbated for languages with limited data available. Prior work using LMs for mKGC relies on pre-training LMs with large amounts of structured data (e.g., Zhou et al. (2022) train on a KG with 52M triples). However, languages on the long tail do not have such datasets available (Joshi et al., 2020). Based on official statistics, only 0.2% of the total entities in Wikidata (Vrandečić and Krötzsch, 2014) have labels in the low-resourced language Amharic.[1] Additionally, most pre-trained LMs do not have good performance for low-resourced languages (Ojo et al., 2024).

We propose mRAKL, a retrieval-augmented sequence-to-sequence generative method for mKGC. mRAKL uses a *retriever* which fetches relevant passages and passes them to a *generator* model which predicts knowledge facts. Moreover, we allow LMs to learn better cross-lingual entity representation leveraging entity parallel textual information from Wikidata (Vrandečić and Krötzsch, 2014). These two approaches greatly alleviate the data scarcity problem. We focus on enriching KGs of two low-resourced languages: Amharic and Tigrinya, two Afro-Semitic languages.

mRAKL is a cross-lingual RAG-based QA system for mKGC (see Figure 1). To use mRAKL, we first reformulate mKGC as a Question-Answering task. We create a QA dataset by transforming each KG

[1] https://www.wikidata.org/wiki/User:Pasleim/Language_statistics_for_items

Findings of the Association for Computational Linguistics: ACL 2025, pages 13072–13089

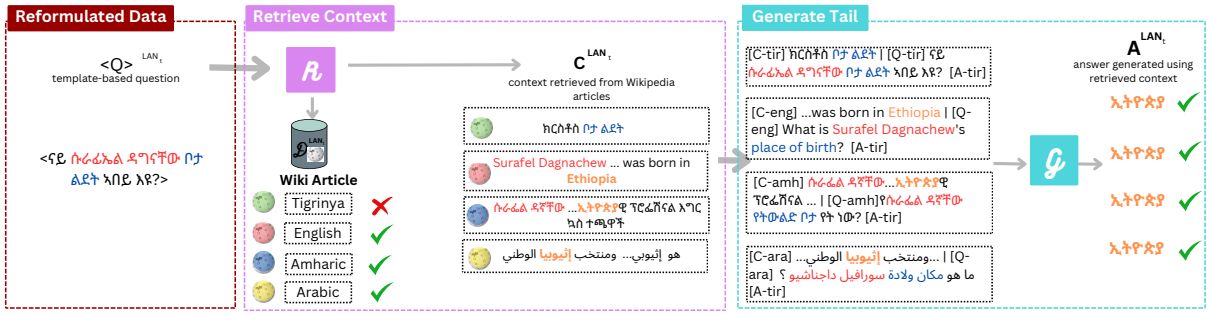

Figure 1: **Inference with mRAKL** In this example our triple is (Surafel Dagnachew, place of birth, Ethiopia). Taking the reformulated question "What is Surafel Dagnachew's place of birth?", the retriever encodes the query and fetches sentences with the highest similarity from the passages available from the Wikipedia articles in each of the languages. These sentences are then passed to the generator as context along with the question (see Appendix C for details on question generation in the four languages).

triple into a question-answer pair: we construct a question text that uses the *head* entity and *relation* from the KG triple and an answer text that uses the *tail* (see Figure 8). For our *generator* model, we finetune a multilingual LM, AfriTeVa (Jude Ogundepo et al., 2022) with cross-lingual entity-centered alignment data. This finetuning allows the model to learn better representations for the entities in the low-resourced languages and unify complementary knowledge across languages. Since data in these languages is small, we train a *retriever* model to further enrich the construction by the *generator*, utilizing monolingual datasets which are more easily available for low-resourced languages compared to structured, labeled datasets (Joshi et al., 2020)

We evaluate our approach and existing approaches on two tasks: probing parametric knowledge of pre-trained LMs for low-resourced languages (§4.2) and monolingual and cross-lingual link prediction (§4.4). We also perform ablation studies showing the benefit of RAG for mKGC where structured data is constrained (§A.2). Overall, we find that mRAKL outperforms prior approaches on Amharic and Tigrinya by an 8 and 6 percentage point increase and that using cross-lingual context improves over no-context settings. Our contributions can be summarized as follows:

- We contribute a 3.5k triple KG for Tigrinya and a 34k triple KG for Amharic. We also contribute our question templates and generative and retrieval models trained for mKGC (§3.1).[2]
- We propose a novel RAG-based approach to mKGC that retrieves relevant passages from unstructured monolingual data for KG completion (§3.2). We show the benefits of using our method for low-resourced languages (§4.3).
- We propose a novel method for cross-lingual entity linking through cross-lingual link prediction, where given a *head* entity and *relation* in one language, the *tail* is predicted in another language (§4.4).

[2] https://github.com/hhnigatu/mRAKL

2 Related Work

KG Construction Prior to LLMs, KGs were automatically constructed with multi-staged rule-based pipelines typically including information extraction, knowledge fusion, and knowledge graph completion (e.g. Lehmann et al., 2015; Hoffart et al., 2013; Dong et al., 2014). Such systems are expensive to construct and maintain since they demand significant human efforts or only consume structured or semi-structured data which is easy for rule-based systems to deal with (Carlson et al., 2010; Vrandečić and Krötzsch, 2014; Zhong et al., 2023). A recent trend explores ways to extract factual knowledge from LLMs with prompting and fine-tuning methods (e.g. Bosselut et al., 2019; Jiang et al., 2020; Saxena et al., 2022a; Kassner et al., 2021; Zhou et al., 2022; Song et al., 2023) both in monolingual and multilingual settings which addresses the drawback of traditional approaches. However, such approaches fall short in dealing with low-resourced languages when data is rarely seen in the training phase. mRAKL employs RAG based approach to alleviate data scarcity.

A number of multilingual KG embedding-based approaches have been proposed to tackle the cross-lingual knowledge alignment and KG completion problems (e.g Chen et al., 2021, 2017; Chakrabarti et al., 2022). The key idea of these approaches is to align the knowledge across KGs in different languages into a unified embedding space so that the link prediction can leverage the complementary knowledge (Chen et al., 2020; Sun et al., 2020). Nevertheless, these approaches assume a closed-world framework which cannot take open-world information and natural language knowledge into consideration. Our system integrates the best of the worlds by combining knowledge from multilingual open-domain text through RAG and multilingual KGs through cross-entity alignment.

Retrieval Augmented Generation RAG augments LLMs with non-parametric memories from one or more external data sources. RAG has attracted significant attention recently because it addresses several critical limitations of LLMs (e.g. Guu et al., 2020; Gao et al., 2023; Lewis et al., 2020); for example, it is expensive to correct outdated, erroneous facts within LLMs through fine-tuning or re-training. RAG offers a way to circumvent this issue by providing up-to-date information to a pre-trained model during generation. Furthermore, RAG enables LLMs to focus on generalization and reasoning which reduces the size of LLMs needed to achieve similar performance (Asai et al., 2023). Given that low-resource languages occupy the long tail of available knowledge during the training phase of LLMs, RAG offers a way to boost the performance for mKGC by retrieving new, open-domain information instead of inferring knowledge only from multilingual KGs. While several research works focus on augmenting LLMs with KGs (Pan et al., 2024; Yang et al., 2024), to the best of our knowledge mRAKL is the first work to explore how RAG with LLMs can help construct and complete KGs in a low-resourced setting.

3 Proposed Method

Our proposed method is to reformulate mKGC as a cross-lingual question-answering task and use an RAG-based QA system for completion. We first convert the *(head, relation, tail)* triples into a <*question, answer*> format where the question includes the *head* and *relation* and the answer is the *tail* (§3.1). Using a *retriever* model, we extract context from monolingual unlabeled datasets. We then use the *generator* model to predict the tail entity, given the extracted context and the question (Figure 1). Below we first give background on our languages of focus, then detail our data preparation steps (§3.1) and then present mRAKL (§3.2).

Languages of Study Our main languages of study are Tigrinya and Amharic, Afro-Semitic languages that use the Ge'ez script. The languages are considered low-resourced in that there are limited tools and data available to build language technologies for them (Yimam et al., 2020; Gaim et al., 2023). Tigrinya is one of the official languages of Ethiopia and Eritrea and is spoken by 9.7 million people[3] in total across the two countries and their diasporas. Amharic is one of the official languages of Ethiopia, spoken by over 33.7 million people as a first language and 25.1 million as a second language (Basha et al., 2023). To enrich the data for these languages, we chose two transfer languages: Arabic and English. We selected Arabic as a transfer language because (1) it is in the same language family as Tigrinya and Amharic (motivated by Ogunremi et al. (2023))and (2) we hypothesize there are cultural and geographic ties that would make the information more culturally relevant (motivated by Zhou et al. (2022)). We select English as a second transfer language because of the abundance of resources in the language.

3.1 Data Preparation

In this section, we will describe our data collection process by detailing the steps we took for Tigrinya, one of our target languages. We will then give the statistics of the data for all of the languages in our experiments in Table 2. The process when Amharic is the target language is exactly the same as detailed below for Tigrinya. When working on one of our target languages, we use the other languages as transfer languages; for instance, when Tigrinya is the target language, Amharic, English, and Arabic are transfer languages.

Relations: We first collected textual representations of the relations in Tigrinya from the Wikidata Property Explorer.[4] This resulted in 96 relations for Tigrinya. We then added 24 relations that have textual representations in Amharic but not in Tigrinya by manually translating the textual representations; this resulted in 120 relations in total.

[3]https://www.ethnologue.com/language/tir/
[4]https://prop-explorer.toolforge.org/

KG Extraction: We extracted the Tigrinya KG from Wikidata using *simple-wikidata-db*.[5] For each entity in Wikidata with a corresponding Wikipedia article title in the Tigrinya Wikipedia, we keep the triples from Wikidata that have the entity as a head or tail. We then filtered through the KG to keep the triples with the relations from our set of 120 relations described above. We extracted a KG with 3.5k edges and 272 unique entities.

Template-Based Reformulation: We then manually prepared question templates for each of the 120 relations[6]. Then, for each triple in the KG, we plug in the textual representation of the head entity into the template and use the textual representation of the tail entity as an answer, resulting in a 3.5k question-answer pair dataset. Figure 8 shows an example of how we use our template-based reformulation approach.

Extracting Context Given the *head* and *relation* in the question, the goal of the generator model is to predict the *tail* as the answer. To enhance the ability of the *generator* model in correctly predicting the tail, we use the *retriever* model to extract sentences that will provide context. Since we do not have labeled data, we devised a heuristic to extract context for our question-answer pairs. We extracted the context for each question by searching for the tail entity in the first paragraph of the Tigrinya Wikipedia article associated with the head entity. We then kept a maximum of two sentences that had the tail entity as context. We use our heuristic context extraction method as an (im)perfect retriever setup: while it does not guarantee the context will always be retrieved (for example, when there is no mention of the tail entity in the head entity's Wikipedia article), when it does provide context, the retrieved context will certainly have the tail entity. However, this is not representative of the real-world task when we do not have access to the tail entity to search the Wikipedia article. Hence, we use the (im)perfect retriever setup to provide an upper bound of performance (§A.2).

Data in Transfer Languages: We retrieved the textual representations for the 120 relations we collected as detailed above in Arabic and English from Wikidata using the relation ID. For Amharic, we manually translated the 68 relations that were unique to Tigrinya. Our final set of relations had 120 relations in the four languages of study. We then manually prepared template questions for each of the 120 relations in Amharic, English, and Arabic. Table 1 gives details of the final dataset and Table 2 gives statistics on the coverage of each KG by the transfer languages. We then got the labels for the head and tail entities in the Tigrinya KG from Wikidata in Amharic, English, and Arabic; extracting the 3.5k triples in our Tigrinya KG from the three transfer languages' KGs. Once we had the final dataset, we split it to train, evaluation and test sets with an 8:1:1 ratio. We use the evaluation set for hyperparameter tuning and report results on the test set which is unseen during training. We then used the same strategy as described for Tigrinya to perform the template-based reformulation and context extraction for Amharic (see Appendix C).

KG	Triples	Head	Tail
Tigrinya	3.5k	244	170
Amharic	34k	8568	5058

Table 1: Details on size of KGs in the two target languages.

		Tigrinya KG		Amharic KG	
Language	Wiki	Head	Tail	Head	Tail
Amharic	14.04K	79.50	86.47	100	100
Arabic	1.23M	95.49	99.41	79.56	94.36
English	6.84M	100	100	90.40	98.39
Tigrinya	506	100	100	3.60	4.03

Table 2: Percentage of the head and tail entities in each of the target language KGs with textual representations in each of the transfer languages.

3.2 mRAKL

Our proposed setup involves a *retriever* that extracts the necessary context and passes it to the *generator* which, given context and a template question (i.e. a question with the relation and head entity), generates the answer (i.e. the tail). Figure 1 shows our complete setup.

Input Representation For what follows, we will use t to refer to the target language. In the input sequence notation below, $\texttt{LAN}_t$ represents the three-letter ISO language code for language t. We use `Q` to represent the question text, `A` to represent the

[5] https://github.com/neelguha/simple-wikidata-db/tree/main

[6] See Appendix C for translation and manual question preparation details.

answer text, and C to represent the context text. Our input sequences also use the special tokens [C-LAN], [Q-LAN], and [A-LAN] to indicate the start of a context, question, and answer respectively[7]. We use the '?' symbol to mark the end of a question[8] and 'l' to mark the end of a context. For the retriever, we pass the questions Q as queries and retrieve context C for each question. We concatenate the retrieved context C and the question Q as follows:

[C-LAN_t]C | [Q-LAN_t]Q? [A-LAN_t]
$\forall\, t \epsilon \{Tigrinya, Amharic, English, Arabic\}$

The last element of the input sequence is the [A-LAN] token, which indicates in which language the model should generate answers. Hence, the retriever's task is, given Q with the *head* and *relation*, retrieve C that ideally includes the answer A which is the *tail* entity. The generator's task is, given C which contains the *tail* entity and Q with the *head* and *relation*, predict A which is the tail entity.

Training: To train mRAKL for link prediction, we prepare our template-based question-answering data as detailed in §3.1. For the triple (head, relation, tail), the question has the head and relation, and the answer is the tail (Figure 8). For the *retriever* model, we use BM25 (Robertson and Zaragoza, 2009) and LaBSE(Feng et al., 2022) For the *generator* model, we finetune the AfriTeVa-base model with LoRA(Hu et al., 2022). During the *generator* training, the model is trained with cross-entropy loss. Similar to (Saxena et al., 2022b), we do not use explicit negative sampling. We provide both *generator* and *retriever* model training details in Appendix B.

Inference: Given a query (head, relation, ?), we first convert it to a question-answer format (§3.1). We then feed the question, which has the *head* and *relation*, to our *retriever* to extract context from monolingual passages. We then feed the extracted context, which ideally would have the tail entity, to the *generator*. The *generator* takes the context and question as input and produces a probability distribution over all tokens. We use beam-search during decoding with a beam size of 10 and take the top n tokens (where n ϵ {1,3,10}) .

Given we are operating in a low-resourced context, we hypothesize that our RAG-based approach will improve performance over a generator-only approach by allowing the LM to learn the tail entity from an extracted context. Additionally, the multilingual setting allows for cross-lingual entity linking, i.e given a *head* and *relation* in one language, predicting the *tail* in another language. This cross-lingual entity alignment enriches the dataset as well as the learned knowledge representation. Further, the modularity of the RAG pipeline allows us to improve the *retriever* and *generator* models separately; hence, we can utilize unlabeled, monolingual data which is easier to acquire than labeled and structured data (Joshi et al., 2020), to perform mKGC.

4 Experiments and Results

In this section, we evaluate our proposed method with two tasks: (1) Parametric Knowledge Probing, which is a way to test knowledge representations in an LM's learned embeddings (§4.2) and (2) Link Prediction, which is a standard task for evaluating KG completion and construction. For Link Prediction, we evaluate by (1) comparing our method with that of prior work approaches (§4.3) and (2) by looking at cross-lingual link prediction, where we predict the tail entity in one language given a head and relation entity in another language (§4.4). We also perform additional analysis with our (im)perfect retriever.

Overall, we find that our approach outperforms prior methods by over a 4.9 percentage point increase and that providing context improves over a no-context setting by 6.7 and 12.22 percentage point increase. We also find that multilingual and cross-lingual approaches are more beneficial for Tigrinya, where the data is limited and the transfer languages cover the majority of the entities in the target language (see §C). All results reported are based on a single inference run.

4.1 Experimental Setup

Below, we describe the different experimental settings we tried for our retriever and generator.

Retriever: We use the Wikipedia articles from (Foundation) and for all languages use the 2024-07-01 version [9]. We experiment with the following retrieval setups:

- **(Im)perfect Retriever:** As described in §3.1, we use our (im)perfect retriever to provide an

[7]This is inspired by prior work from (Zhou et al., 2022)
[8]Note that we use ؟ for Arabic questions.
[9]We access the data from https://huggingface.co/datasets/olm/wikipedia

approximate upper bound for how well our system will perform with a good retriever.

- **BM25**: We used the implementation by Lù (2024) for the BM25 indexes. We indexed the full Wikipedia articles of all head entities for retrieval. We built monolingual indexes for each of the four languages. (see Appendix B for details.)

- **LaBSE**: We use the LaBSE model (Feng et al., 2022) as our retriever. LaBSE includes Amharic, Arabic, and English but not Tigrinya. We finetune the LaBSE model with contrastive loss by creating training dataset using our (im)perfect retriever (see Appendix B for details.)

Generator: We experiment with four different setups for training the generator model. Since we are interested in performance both on Tigrinya and Amharic, we use each of the four setups for each of those two target languages.

- No Context—question-answer pairs only without context. In this case, the input to our model is: `[Q-`LAN_t`]Q? [A-`LAN_t`]`.

- Monolingual Self-Context—where question-answer pairs have context in the target language only; the input to our model is: `[C-`LAN_t`]C |[Q-`LAN_t`]Q? [A-`LAN_t`]`.

- Multilingual Self-Context—where question-answer pairs along with the context are in the same language, for all four languages; the input to our model is: `[C-`$LAN_{t'}$`]C |[Q-`$LAN_{t'}$`]Q? [A-`$LAN_{t'}$`]` $\forall\ t' \epsilon \{Tigrinya, Amharic, English, Arabic\}$.

- Cross-Lingual Context—the context and the question are in the same language but the answer may be in any of the four languages; model input is: `[C-`$LAN_{t'}$`]C |[Q-`$LAN_{t'}$`]Q? [A-`$LAN_{t''}$`]` $\forall\ t', t'' \epsilon \{Tigrinya, Amharic, English, Arabic\}$.

4.2 Probing Parametric Knowledge

Task Description Probing parametric Knowledge of LMs involves using prompts to get predictions from a pre-trained LM(Petroni et al., 2019). We perform this probing task in zero-shot on four pre-trained LMs. Testing the models in zero-shot gives us insight into the parametric knowledge (Yu et al., 2024) of the models for these languages. As our prompts, we use the template-based questions from our test set as described in §3.1.

Language →		Tigrinya	Amharic
Zero-Shot	mT5*	-	0.49
	AfriTeVa †	0.22	0.61
	Aya*	0.67	1.52
	GPT-4	**2.23**	**5.83**
Finetuned	mT5	2.01	23.32
	AfriTeva	**5.13**	**29.15**

Table 3: **Zero-shot and finetuned model H@1 results for the no-context setup.** * indicates model does not include Tigrinya but includes Amharic. † indicates model includes both Tigrinya and Amharic. We omit the zero-shot performance of mT5 on Tigrinya as it was worse than AfriTeVa and the language is unseen for the model.

Models in Comparison We compare GPT-4o (OpenAI et al., 2024), Aya(Üstün et al., 2024), mT5 (Xue et al., 2021), and AfriTeVa (Oladipo et al., 2023) . For mT5 and AfriTeva, we use the base models. AfriTeVa includes both target languages in its pre-training while Aya and mT5 include Amharic but not Tigrinya. Arabic and English are included in all models.

Metric: We use H@1 as our metric. We count it as a hit if the prediction contains the tail entity; for instance, if the target is "Addis Ababa" and the model prediction is "It is Addis Ababa," we count it as a hit[10].

Results and Discussion All models perform poorly in both languages in a zero-shot setting, never surpassing 6% (Table 3). We find that the large, generative models, GPT-4o and Aya, outperform the Seq2Seq models with GPT-4o having the highest performance for zero-shot. For Tigrinya, the mT5 model did not produce meaningful predictions in the target language; we hypothesize this is due to the language being unseen for the model and omit the results from the table. We find that AfriTeVa outperforms mT5 on our dataset both before and after fine-tuning. Based on these results, we use AfriTeVa as our base model.

[10]For details on how we attempted to constrain model outputs to predict the tail entity only, refer to Appendix C.

	Tigrinya KG		Amharic KG	
	H@1	H@10	H@1	H@10
KGT5-No-Context	6.91	28.57	32.58	52.57
KGT5-Description	5.8	23.44	32.91	43.32
KGT5-One-Hop	4.46	24.33	28.83	48.17
(ours) No-Context	5.13	26.11	29.15	54.81
(ours) Self-Context	**11.83**	**34.59**	**41.37**	**61.87**

Table 4: Comparison of our proposed method with that of prior work for low-resourced languages where there is limited structured data in a Monolingual setting.

4.3 Closed vs Open Domain Link Prediction

Task Description Link prediction in KG Construction literature (Zhou et al., 2022) refers to the task of predicting a tail entity given a head entity and relation. In this section, we study the impact of RAG for `mRAKL`. For a baseline comparison, we adopt the setting of KGT5 (Saxena et al., 2022a) and KGT5-context (Kochsiek et al., 2023). KGT5 is a monolingual model trained on the Wikidata5M (Wang et al., 2021) dataset. We use their verbalization scheme for link prediction: given a triple (*head, relation, tail*), the input to the model is "`predict tail: head | relation`" and the expected output is the tail entity. We train our base model, AfriTeVa-base, with this scheme on our dataset and compare it to our No-Context setup. For comparison with our RAG-based system, we adopt the setup from Kochsiek et al. (2023) where the model is given additional context during training by (1) appending the description of each entity from Wikipedia (KGT5-Description) and (2) appending the entities directly connected to the head entity (KGT5-One-Hop).

Metric: We use H@1, H@3, and H@10 to denote Hit at the models' top 1, top 3, and top 10 predictions. Hit is counted only if the prediction is an Exact Match (EM) of the tail entity.

Results and Discussion As Table 4 shows, we find that our proposed method with context outperforms the adopted approach from prior work by a 4.92 and 8.79 increase in percentage points for Tigrinya and Amharic respectively for H@1[11]. While in the No-context case, the KGT5 approach outperforms our no-context setting by 1.78 and 3.43 percentage points, we see that adding the structured context (i.e descriptions of entities from Wikidata and one-hop connections) degrades the performance for our target languages. Table 9 shows that structured context required by prior work is not readily avaiable for these languages and when it is avaiable, it rarely contains the tail entity. We hypothesize this limited availability of descriptions and one-hop connections in the low-resourced languages leads to closed-domain link prediction methods being limited for low-resourced languages. Hence, we find that our approach of using unstructured data for retrieving context is a better approach for low-resourced languages.

[11]Performance gain calculated by taking the difference between best-performing methods for each work.

4.4 Cross-Lingual Link Prediction

Task Description In addition to the monolingual link prediction task in Section 4.3, we propose and perform a cross-lingual link prediction task where the head and relation are in one language and the tail is in another language (see §4.1). Here, we do not compare to KGT5 or KGT5-Context as the approach is for monolingual settings. Instead, we compare two retrievers and a no-context setting.

Results and Discussion Table 5 shows performance for the cross-lingual link prediction task. We find that using the LaBSE retriever does not improve performance over the BM25 retriever. However, both retriever options show a gain in performance as compared to the no-context setup. We also observe that the Hit@10 with Amharic as a context is highest compared with other languages as context depending on the available context and the overlap with the target language. We provide detailed analysis in Section A.2.

4.5 Additional Analysis

Effects of Multilingual Context As Table 6 shows, with the (im)perfect retriever, Multilingual Self-Context—using multiple languages but matching the language of any given prompt's context, question, and answer—improves performance over training only with the target language data. The performance boost is especially pronounced for Tigrinya, where adding context results in a 4.69 percentage point increase compared to the Monolingual Self-Context setting. As Figure 4 shows, for the Tigrinya KG, we observe that Arabic and English each provide context for 25% of the dataset while the Amharic and Tigrinya each provide context for less than 10% of the test data. However, we see that H@1 is the highest when Amharic provides

Target lang. →		Tigrinya					Amharic				
Context lang. →		Amh	Ara	Eng	Tir	Avg.	Amh	Ara	Eng	Tir	Avg.
H@1	No-Context	11.64	12.08	14.06	14.06	12.97	35.89	31.51	36.63	8.29	33.12
	LaBSE	12.10	10.29	13.17	13.62	12.30	34.27	30.29	36.07	10.49	32.19
	BM25	**15.75**	**12.30**	**14.73**	**13.84**	**14.15**	**38.52**	**33.58**	**38.22**	**11.17**	**35.27**
H@3	No-Context	**22.60**	**21.48**	22.77	22.32	22.29	44.70	40.41	45.69	**17.60**	42.06
	LaBSE	21.19	18.12	20.76	20.38	22.53	43.80	39.16	44.76	17.26	41.08
	BM25	21.92	21.70	**23.21**	**23.44**	**22.57**	**46.32**	**41.34**	**46.62**	16.75	**43.11**
H@10	No-Context	**39.72**	36.91	38.83	38.16	38.40	54.65	49.86	**56.04**	29.95	52.14
	LaBSE	39.50	36.02	36.38	37.95	37.45	52.48	48.28	53.52	**30.80**	50.22
	BM25	37.67	**35.35**	**38.17**	**38.62**	**37.45**	**54.97**	**50.25**	55.44	30.12	**52.18**

Table 5: Cross-Lingual Link Prediction results broken down by the context language.

	Tigrinya			Amharic		
	None	Target	Avg.	None	Target	Avg.
Mono	7.34	54.76	11.83	**30.70**	79.47	42.92
Multi	**9.60**	**69.05**	15.18	30.53	**80.85**	**43.21**

Table 6: Breakdown of H@1 results by the availability of context in the target language with the (im)perfect retriever. Results for Monolingual-Self Context (Mono) and Multilingual-Self Context (Multi) settings.

context. As demonstrated in Table 2, 86.47% of the tail entities in Tigrinya have corresponding labels in Amharic. We further investigated the overlap of tail entities between the two languages that are spelled the same–i.e would have the same learned representation in the model–and found that 35.88% of the tail entities in Amharic and Tigrinya share the same spelling (Figure 11). This partially explains the boost we observe for using Amharic context. Similarly, for Amharic KG, Figure 5 shows that Arabic provides context for 25% of the dataset while English provides context for 30%. Tigrinya provides context for less than 1% of the data. Amharic, the target language, provides context for 12% of the test set. We see that performance is highest when Amharic provides context. In both Amharic and Tigrinya KGs, while questions without context dominate the test set (43% and 34% respectively), performance on those questions is the lowest, showing the advantage of cross-lingual context.

Qualitative Analysis We qualitatively looked at model predictions to get insights into what the models learned. In Figure 6, we show an example from the BM25 retriever where the tail entity is in the retrieved context for Arabic but not in the context for Amharic (our target language) or English. The Amharic context is unrelated to the query, "What is Blue an instance of?"; the English context is related in that it talks about a blue dye, even though it does not use the term 'color'; the Arabic context talks about the blue, red, and green colors of the Azerbaijan flag, providing context that our head entity "blue" is a type of color. In this example, we see that our system benefits from the cross-lingual transfer, where the entity names in the three languages are aligned and the model can correctly predict the head entity in all three languages. This is an instance of how our cross-lingual entity alignment (Sec. 4.4) works. While prior work (e.g. Zhou et al., 2022) relies on explicitly aligning entities in different languages, in our approach, the alignment is done through the cross-lingual context provided to the generative model. Refer to Appendix A for additional qualitative analysis.

Further, we looked at cases where the transfer language is English as compared to Arabic. Supporting our hypothesis, we find that Arabic helps with some queries that are regionally specific to the target languages: for instance, queries like "What is Afar's writing script?" where the English provided context outputs an incorrect prediction. Further, we observe that the Arabic context helps for queries related to Middle Eastern and Asian contexts such as "What is Arwad's country?" and "What is Gwadar Port's country?" On the other hand, queries like "What country is Madrid the capital of?" were correctly predicted as "Spain" when English context was provided, but "Afghanistan" when Arabic context was provided. This suggests that the English context provides support for more Western

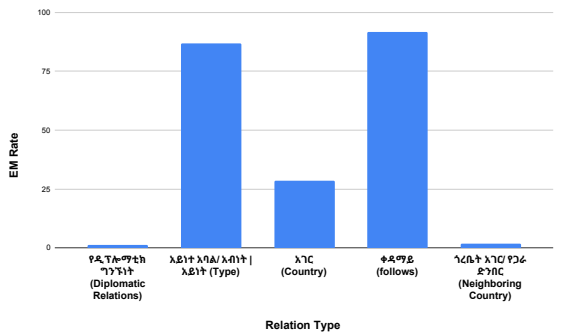

Figure 2: Percentage of triples with the given relation that had correct tail predictions for top 5 most frequent relations in Amharic test set.

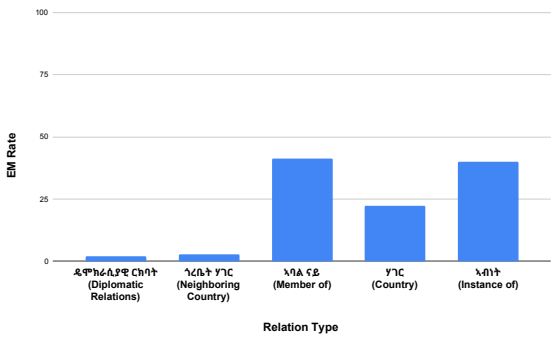

Figure 3: Percentage of triples with the given relation that had correct tail predictions for top 5 most frequent relations in Tigrinya test set.

topics while the Arabic context provides support for more culturally/regionally specific and Middle East/Asian queries. Hence, there is potential for future work to explore how to best select transfer languages that support diverse cultures.

Bias in Data Prior work has demonstrated that there are societal biases in Wikipedia across languages (Samir et al., 2024). Specifically in relation to our languages of study, prior work has shown that Wikipedia articles might contain "harmful and abusive content."(Nigatu et al., 2024). In light of these prior works, it is crucial to interrogate our findings and data. In this work, we specifically looked at gaps between the KGs of each of the target languages. We find that entities that exist in Tigrinya KG but not in Amharic KG are mostly regions in Eritrea like the Northern Red Sea Region and Gash-Barka Region. Hence, using Amharic as a transfer language provides data for entities shared in common by Ethiopia and Eritrea but lacks representation for entities that are exclusively related to Eritrea. Similarly, we find that what is not covered by the transfer languages for the Amharic KG is mostly related to famous Ethiopians. Therefore, while transfer languages can help provide context for entities that are shared across the languages, there are gaps in transfer language KGs for entities that are specific to the target languages. We give more details on this in Appendix C.

Analysis by Relation Type To understand which relations were being correctly linked, we looked at the triples for which our system correctly predicted the tail. Specifically, we looked at the distribution of the relations for which the triples were correctly completed. We used the BM25 retriever model setup for this analysis. For both target languages, "diplomatic relations" is the most frequent relation type; i.e majority of the triples in the test set have the relation "diplomatic relations.". In Figures 2 and 3, we show the distribution of the top 5 most frequent relations in the Amharic and Tigrinya test set respectively, along with the percentage of triples with each relation that were correctly predicted by the mRAKL system. In both target language cases, triples with "diplomatic relations" and "neighboring country" have the least percentage of correctly predicted tails. This could be because they have a many-to-many relationship. On the other hand, we observe that Amharic has higher percentage of triples with top 5 most frequent relations correctly predicted, indicating the benefits of training from more data as the model will have access to additional context.

5 Conclusion

We propose mRAKL, a RAG-based approach for mKGC in low-resourced language settings. We have shown that a RAG-based finetuned LM can retrieve facts that help with mKGC. In addition, our cross-lingual entity alignment technique combines complementary knowledge across languages and increases the available corpus for RAG. Our experimental results demonstrate that mRAKL increases accuracy by up to 4.92 and 8.79 percentage points for Tigrinya and Amharic, respectively, compared with baselines. Our approach represents a step towards alleviating the cultural knowledge scarcity that LLMs typically display during pre-training.

Limitations

Our work has several limitations: First, when using the (im)perfect retriever, the extracted context

may not be the exact context needed (see §3 and Appendix B). However, it does correctly link the head and relation to the tail entity. With the limited labeled data available for our languages of focus, the heuristic method was the best option we could come up with to provide an upper bound on what mRAKL can achieve. Additionally, we focused our efforts on two low-resourced languages. We do not make claims about the efficacy of our method for other low-resourced languages; we only had resources (human and computational) sufficient to work on the two languages. However, low-resourced languages differ in the available resources; some languages may have even more limited monolingual data. To account for this, we provide results with and without context. Additionally, the two languages are related and use the same script. Future work can explore to what extent our approach can be extended to other low-resourced languages. We also relied on manual effort to construct the templates for each of the languages. While this requires human labor, we were interested in providing high-quality data for these languages. Future work could explore using automated methods (e.g using machine translation). Additionally, our work does not look into co-reference resolution, i.e, resolving multiple names of an entity. Currently, our work relies on the generative model to implicitly resolve multiple entity names. We will explore more explicit co-reference resolution techniques in future work.

Acknowledgments

We would like to thank members and friends of PLAIT, EPIC and Canny Lab for their feedback on this work. We also want to thank Nuredin Ali and Negasi Haile for their help in reviewing the template questions for Tigrinya and Mustafa Abuzahriyeh, Habiba Geweifal, and Anisa Mohammed for their help with the Arabic template questions. We thank Hailay Teklehaymanot for early conversations about this work. We thank the reviewers for their valuable feedback on our paper.

References

Akari Asai, Sewon Min, Zexuan Zhong, and Danqi Chen. 2023. Acl 2023 tutorial: Retrieval-based language models and applications. *ACL 2023*.

Shaik Johny Basha, Duggineni Veeraiah, Boddu Venkat Charan, Wiltrud Sahithi Joyce Yeddu, and Devalla Ganesh Babu. 2023. Detection and comparative analysis of handwritten words of amharic language to english using cnn-based frameworks. In *2023 International Conference on Inventive Computation Technologies (ICICT)*, pages 422–427.

Antoine Bosselut, Hannah Rashkin, Maarten Sap, Chaitanya Malaviya, Asli Celikyilmaz, and Yejin Choi. 2019. Comet: Commonsense transformers for automatic knowledge graph construction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4762–4779.

Andrew Carlson, Justin Betteridge, Bryan Kisiel, Burr Settles, Estevam Hruschka, and Tom Mitchell. 2010. Toward an architecture for never-ending language learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 24, pages 1306–1313.

Soumen Chakrabarti, Harkanwar Singh, Shubham Lohiya, Prachi Jain, et al. 2022. Joint completion and alignment of multilingual knowledge graphs. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, page 11922—11938.

Muhao Chen, Weijia Shi, Ben Zhou, and Dan Roth. 2021. Cross-lingual entity alignment with incidental supervision. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 645–658.

Muhao Chen, Yingtao Tian, Mohan Yang, and Carlo Zaniolo. 2017. Multilingual knowledge graph embeddings for cross-lingual knowledge alignment. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence*. International Joint Conferences on Artificial Intelligence Organization.

Xuelu Chen, Muhao Chen, Changjun Fan, Ankith Uppunda, Yizhou Sun, and Carlo Zaniolo. 2020. Multilingual knowledge graph completion via ensemble knowledge transfer. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3227–3238, Online. Association for Computational Linguistics.

Xin Dong, Evgeniy Gabrilovich, Geremy Heitz, Wilko Horn, Ni Lao, Kevin Murphy, Thomas Strohmann, Shaohua Sun, and Wei Zhang. 2014. Knowledge vault: A web-scale approach to probabilistic knowledge fusion. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 601–610.

Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. 2022. Language-agnostic BERT sentence embedding. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 878–891, Dublin, Ireland. Association for Computational Linguistics.

Wikimedia Foundation. Wikimedia downloads.

Fitsum Gaim, Wonsuk Yang, Hancheol Park, and Jong Park. 2023. Question-answering in a low-resourced language: Benchmark dataset and models for Tigrinya. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11857–11870, Toronto, Canada. Association for Computational Linguistics.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, and Haofen Wang. 2023. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. 2020. Realm: retrieval-augmented language model pre-training. In *Proceedings of the 37th International Conference on Machine Learning*, pages 3929–3938.

Johannes Hoffart, Fabian M Suchanek, Klaus Berberich, and Gerhard Weikum. 2013. Yago2: A spatially and temporally enhanced knowledge base from wikipedia. *Artificial intelligence*, 194:28–61.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Xiao Huang, Jingyuan Zhang, Dingcheng Li, and Ping Li. 2019. Knowledge graph embedding based question answering. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, WSDM '19, page 105–113, New York, NY, USA. Association for Computing Machinery.

Jinhao Jiang, Kun Zhou, Xin Zhao, and Ji-Rong Wen. 2023. Unikgqa: Unified retrieval and reasoning for solving multi-hop question answering over knowledge graph. In *The Eleventh International Conference on Learning Representations*.

Zhengbao Jiang, Antonios Anastasopoulos, Jun Araki, Haibo Ding, and Graham Neubig. 2020. X-factr: Multilingual factual knowledge retrieval from pretrained language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5943–5959.

Pratik Joshi, Sebastin Santy, Amar Budhiraja, Kalika Bali, and Monojit Choudhury. 2020. The state and fate of linguistic diversity and inclusion in the NLP world. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6282–6293, Online. Association for Computational Linguistics.

Odunayo Jude Ogundepo, Akintunde Oladipo, Mofetoluwa Adeyemi, Kelechi Ogueji, and Jimmy Lin. 2022. AfriTeVA: Extending ?small data? pretraining approaches to sequence-to-sequence models. In *Proceedings of the Third Workshop on Deep Learning for Low-Resource Natural Language Processing*, pages 126–135, Hybrid. Association for Computational Linguistics.

Nora Kassner, Philipp Dufter, and Hinrich Schütze. 2021. Multilingual lama: Investigating knowledge in multilingual pretrained language models. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3250–3258.

Adrian Kochsiek, Apoorv Saxena, Inderjeet Nair, and Rainer Gemulla. 2023. Friendly neighbors: Contextualized sequence-to-sequence link prediction. In *Proceedings of the 8th Workshop on Representation Learning for NLP (RepL4NLP 2023)*, pages 131–138, Toronto, Canada. Association for Computational Linguistics.

Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick Van Kleef, Sören Auer, et al. 2015. Dbpedia–a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic web*, 6(2):167–195.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474.

Xing Han Lù. 2024. Bm25s: Orders of magnitude faster lexical search via eager sparse scoring.

Hellina Hailu Nigatu, John Canny, and Sarah E. Chasins. 2024. Low-resourced languages and online knowledge repositories: A need-finding study. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*, CHI '24, New York, NY, USA. Association for Computing Machinery.

Tolulope Ogunremi, Dan Jurafsky, and Christopher Manning. 2023. Mini but mighty: Efficient multilingual pretraining with linguistically-informed data selection. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 1251–1266, Dubrovnik, Croatia. Association for Computational Linguistics.

Jessica Ojo, Kelechi Ogueji, Pontus Stenetorp, and David Ifeoluwa Adelani. 2024. How good are large language models on african languages?

Akintunde Oladipo, Mofetoluwa Adeyemi, Orevaoghene Ahia, Abraham Toluwase Owodunni, Odunayo Ogundepo, David Ifeoluwa Adelani, and Jimmy Lin. 2023. Better quality pre-training data and t5 models for african languages. In *The 2023 Conference on Empirical Methods in Natural Language Processing*.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin,

Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, ukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, ukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. 2024. Gpt-4 technical report.

Shirui Pan, Linhao Luo, Yufei Wang, Chen Chen, Jiapu Wang, and Xindong Wu. 2024. Unifying large language models and knowledge graphs: A roadmap. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*.

Heiko Paulheim. 2018. How much is a triple? estimating the cost of knowledge graph creation.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language models as knowledge bases? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2463–2473, Hong Kong, China. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Ridho Reinanda, Edgar Meij, and Maarten de Rijke. 2020. Knowledge graphs: An information retrieval perspective. *Found. Trends Inf. Retr.*, 14(4):289–444.

Stephen Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389.

Farhan Samir, Chan Young Park, Anjalie Field, Vered Shwartz, and Yulia Tsvetkov. 2024. Locating information gaps and narrative inconsistencies across languages: A case study of LGBT people portrayals on Wikipedia. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 6747–6762, Miami, Florida, USA. Association for Computational Linguistics.

Apoorv Saxena, Adrian Kochsiek, and Rainer Gemulla. 2022a. Sequence-to-sequence knowledge graph completion and question answering. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2814–2828.

Apoorv Saxena, Adrian Kochsiek, and Rainer Gemulla. 2022b. Sequence-to-sequence knowledge graph completion and question answering. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2814–2828, Dublin, Ireland. Association for Computational Linguistics.

Ran Song, Shizhu He, Shengxiang Gao, Li Cai, Kang Liu, YU Zhengtao, and Jun Zhao. 2023. Multilingual knowledge graph completion from pretrained language models with knowledge constraints. In *The 61st Annual Meeting Of The Association For Computational Linguistics*.

Zequn Sun, Chengming Wang, Wei Hu, Muhao Chen, Jian Dai, Wei Zhang, and Yuzhong Qu. 2020. Knowledge graph alignment network with gated multi-hop neighborhood aggregation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 222–229.

Shiyu Tian, Yangyang Luo, Tianze Xu, Caixia Yuan, Huixing Jiang, Chen Wei, and Xiaojie Wang. 2024. Kg-adapter: Enabling knowledge graph integration in large language models through parameter-efficient fine-tuning. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 3813–3828.

Ahmet Üstün, Viraat Aryabumi, Zheng Yong, Wei-Yin Ko, Daniel D'souza, Gbemileke Onilude, Neel Bhandari, Shivalika Singh, Hui-Lee Ooi, Amr Kayid, Freddie Vargus, Phil Blunsom, Shayne Longpre, Niklas Muennighoff, Marzieh Fadaee, Julia Kreutzer, and Sara Hooker. 2024. Aya model: An instruction finetuned open-access multilingual language model. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15894–15939, Bangkok, Thailand. Association for Computational Linguistics.

Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: a free collaborative knowledgebase. *Communications of the ACM*, 57(10):78–85.

Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: a free collaborative knowledgebase. *Commun. ACM*, 57(10):78–85.

Xiaozhi Wang, Tianyu Gao, Zhaocheng Zhu, Zhengyan Zhang, Zhiyuan Liu, Juanzi Li, and Jian Tang. 2021. KEPLER: A unified model for knowledge embedding and pre-trained language representation. *Transactions of the Association for Computational Linguistics*, 9:176–194.

Yuxiang Wu, Yu Zhao, Baotian Hu, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. 2022. An efficient memory-augmented transformer for knowledge-intensive nlp tasks. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5184–5196.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online. Association for Computational Linguistics.

Linyao Yang, Hongyang Chen, Zhao Li, Xiao Ding, and Xindong Wu. 2024. Give us the facts: Enhancing large language models with knowledge graphs for fact-aware language modeling. *IEEE Transactions on Knowledge and Data Engineering*.

Liang Yao, Chengsheng Mao, and Yuan Luo. 2019. Kg-bert: Bert for knowledge graph completion.

Seid Muhie Yimam, Hizkiel Mitiku Alemayehu, Abinew Ayele, and Chris Biemann. 2020. Exploring Amharic sentiment analysis from social media texts: Building annotation tools and classification models. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 1048–1060, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Haeun Yu, Pepa Atanasova, and Isabelle Augenstein. 2024. Revealing the parametric knowledge of language models: A unified framework for attribution methods. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8173–8186, Bangkok, Thailand. Association for Computational Linguistics.

Lingfeng Zhong, Jia Wu, Qian Li, Hao Peng, and Xindong Wu. 2023. A comprehensive survey on automatic knowledge graph construction. *ACM Computing Surveys*, 56(4):1–62.

Wenxuan Zhou, Fangyu Liu, Ivan Vulić, Nigel Collier, and Muhao Chen. 2022. Prix-LM: Pretraining for multilingual knowledge base construction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5412–5424, Dublin, Ireland. Association for Computational Linguistics.

A More Results

A.1 (im)perfect Retriever

Since we did not have labeled data, we used the (im)perfect retriever as an upper bound for performance. This allowed us to interrogate how well our system performs in an ideal case and to perform a set of ablation studies. To check how well the (im)perfect retriever performs compared to the

Context Language	(im)perfect retriever	LaBSE
Tir	78.05	43.90
Amh	64.19	19.75
Ara	58.87	16.45
Eng	64.51	22.58
No-context in (im)perfect	9.09	12.53

Table 7: Performance comparison of the (im)perfect retriever and the LaBSE retriever broken down by context language.

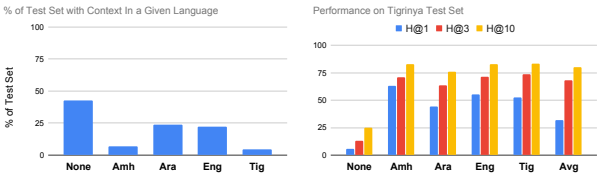

Figure 4: Effects of context from different languages on Hits for Tigrinya.

other retrievers in our experiments, we compare the performance of the (im)perfect retriever with the LaBSE retriever breaking down the results by context language. As Table 7 shows, we find that the (im)perfect retriever outperforms the LaBSE retriever regardless of what language the context is in. However, the (im)perfect retriever, which searches for the tail entity in the Wikipedia article of the head entity, may not always retrieve context (e.g if the head entity Wiki article does not explicitly mention the tail entity). In the cases where the (im)perfect retriever does not fetch context, the LaBSE retriever outperforms. Nonetheless, the (im)perfect retriever provides an upper bound for the cases where context is provided.

A.2 Multilingual Context

In this section, we provide figures and graphs to support the results reported in Sec. . Figure 4 and Figure 5 give a breakdown of performance by context language along with what percentage of the context comes from which language.

A.3 Qualitative Examples

In this section, we provide qualitative evidence that explains our system performance. In Figure 6, we show an example that demonstrates how our system does cross-lingual entity alignment. Only

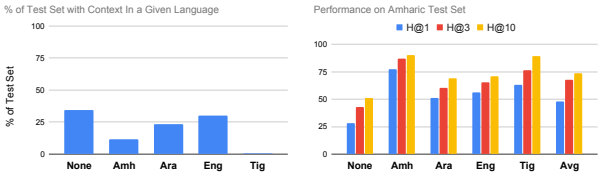

Figure 5: Effects of context from different languages on Hits for Amharic.

Query: ሰማያዊ የምን አይነት ነው? (What is Blue an instance of?)

Tail: ቀለም (Color)

Context Language	Context	Prediction
Amharic	እና እኔም አልሰራ ካሉ ይቅር እንጂ የምን ክስ የምን ጣጣ ነው ብየ ተከራከርኩ በርካታ መጽሔቶችም የምን ጊዜም ታላቁ አርቲስት በማለት ይገልጹታል	ቀለም ✓
Arabic	اللون انتفض القيصر واقفا للاحتجاج وصاح القيصر في عجب ماذا ماذا هيكل نموذج أزرق أحمر أخضر أمر استخدام علم جمهورية أذربيجان ينص على أن لون علم الدولة دقيق	لون ✓
English	That is what happened in this instance For instance he hosted a dinner party where he dyed all the food blue because he claimed there weren t enough blue foods	color ✓

Figure 6: Context from Arabic contains the tail entity but context from Amharic and English does not; however, as a result of cross-lingual alignment, the generative model is correctly able to predict the tail entity in all languages.

the Arabic context includes the target tail entity. However, the generative model is able to predict the tail entity in all three languages, suggesting that it was able to correctly align the entity 'color' in all the languages.

In Figure 7, we observe an example where the Heuristic (im)perfect retriever and the BM25 retrieved the correct context, while the LaBSE retriever did not. Both the (im)perfect retriever and BM25 were able to find a context that included the head, relation, and tail; while the LaBSE model context only has the head entity. The first sentence retrieved by the LaBSE model translates to "Porto-Novo, which was known as Ajashe, was the main location for the Aja government." While the context contains information about the head entity, it does not answer the query "What is Porto-Novo a capital of?" In this case, the prediction of the generative model with the LaBSE model was incorrect. However, since our approach is modular, the retriever can be improved separately which will increase performance.

B Further Details on Methods

Here, we provide details on how we trained our *generator* and *retriever* models. For BM25, we

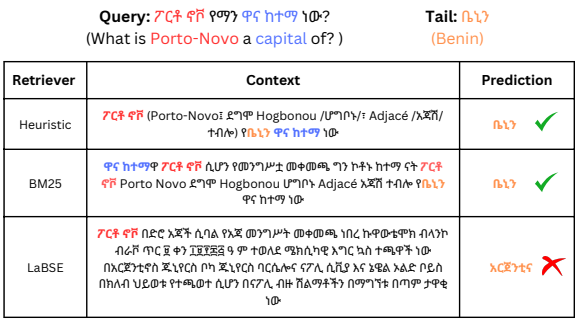

Figure 7: Example where the BM25 and Heuristic retriever were able to fetch context that includes the head, relation, and tail while the LaBSE model context only contains the head entity. As a result, the generative model makes an incorrect prediction.

used the library by Lù (2024) with default settings. Below, we describe our finetuning setup for LaBSE and AfriTeva.

B.1 Finetuning LaBSE

While the LaBSE model (Feng et al., 2022) includes Amharic, Arabic and English, it does not include Tigrinya. Since we do not have labeled data for finetuning, we used the contexts from our (im)perfect retriever to prepare training data for the LaBSE model. The (im)perfect retriever extracts sentences that have the tail entity from the introduction paragraph of the Wikipedia article for the head entity. We call this retriever (im)perfect because:

- It may not always retrieve context. For instance, if the tail entity is not mentioned at all in the introduction paragraph of the Wikipedia article, the (im)perfect retriever will not return anything.
- Some of the sentences it retrieves may not be a direct answer to the query. For example, if the question is "What is Surafel Dagnachew's place of birth?", the retrieved sentence may be "Surafel Dagnachew plays for the football team of Ethiopia.". While it does not directly answer the question, it does include the tail entity.

Using the training data of both Amharic and Tigrinya KG with context retrieved by the (im)perfect retriever, we finetune the LaBSE model. We use contrastive loss during training. Training the model requires an *anchor*, which is the query we are using for retrieval, and *positive* and *negative* examples. The extracted context from the (im)perfect retriever serve as the positive example; i.e the model learns to increase the similarity score between the anchor and this positive example. To walk us through a training step, let us take the triple (Surafel Dagnachew, place of birth, Ethiopia). Once reformulated to question answering format, teh training data point becomes <What is Surafel Dagnachew's place of birth?, Ethiopia>. For the *negative* examples, we first get all the sentences that do not include the tail entity (in this case, Ethiopia). We then prepare three types of negative examples:

- Hard Negative: A sentence that does not have any of the entities or relation (head and tail entities or relation). (e.g "Barack Obama was the president of the United States of America.")
- Head Negative: A sentence that contains neither the tail entity nor the head entity but contains the relation. (e.g "Barack Obama's place of birth is the United States of America.")
- Relation Negative: A sentence that contains neither the tail entity nor the relation but contains the head entity. (e.g "Surafel Dagnachew joined Fasil Kenema in 2018.")

We finetuned the model for 50 epochs with a learning rate of 3e-5 and warm up for the first 15% of the training steps. We accessed the model and conducted our finetuning through SentenceTransformers(Reimers and Gurevych, 2019).

Parameter	Value
training_batch_size	16
eval_batch_size	4
epochs	15 or 30
learning_rate	3e-4
lora_rank	4
lora_dropout	0.01
lora_alpha	32

Table 8: Hyperparamters for training mT5 and AfriTeVa models.

B.2 Model Training Hyperparameters

In Table 8, we give the hyperparameter details for training mT5 and AfriTeVa models. For both models, we used the base model version which has 580M and 428M parameters respectively. Training was done on two Titan RTX GPUs.

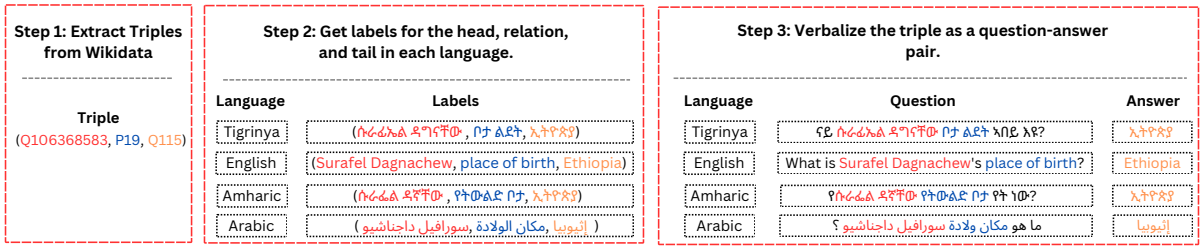

Figure 8: **Reformulating triples into question-answer pairs.** In each step depicted above, we highlight the head entity in red, the relation in blue, and the tail entity in orange. In Step 1, we start with a triple that has entity IDs and Property ID extracted from Wikidata. In Step 2, we get the labels for each of the entity and property IDs in the four languages; this gives us the textual representation of the entities and relations in the different languages. In Step 3, we plug in the head entity into the corresponding template question that has the relation; i.e the head entity *Surafel Dagnachew* is plugged into the template *What is ___'s place of birth?* and the tail entity, *Ethiopia* is the answer the model will learn to predict.

C Data Description

C.1 Manual Data Preparation

For the Amharic data, the first author (L1 Amharic speaker) prepared the template questions manually. For the Tigrinya and English data, the first author (L2 English and L3 Tigrinya speaker) prepared the template questions. For the Tigrinya questions, two L1 speakers checked and corrected any errors. For the Arabic data, two L1 speakers created template questions. For Amharic, Tigrinya, and Arabic, we create questions in both male and female gender when necessary as the three languages are gendered.

C.2 Details on Knowledge Graph

As detailed in Table 2, there is a difference in the percentage of head and tail entities in the target languages that are covered by the transfer languages. Additionally, the number of Wikipedia articles available for each of the four languages of study varies significantly (see Table 2. We took a deeper look at the entities that exist in the target language but do not exist in the transfer language. For the Amharic KG, of the 797 head entities that do not have textual representation in English, 268 have the tail entity "Human" and are names of individuals. We observe that 113 of the individuals are names of famous Ethiopians like Birtukan Dubale or Bahta Gebrehiwot or Ethiopian writers like Mimi Sebhatu and Sheh Tolha Jafar. Those entities are not covered by Tigrinya or Arabic. Of the 3125 head entities in Amharic KG not covered by Arabic, 592 have tail entity "human" and are names of individuals including famous Ethiopians as well as writers like Harold MacGrath and James Sallis which do exist in English KG. In the Tigrinya KG, of the 16 head entities that do not exist in Arabic KG, 7 of them are covered by Amharic KG and include traditional musical instruments of Ethiopia and Eritrea and locations in Ethiopia.

We also looked at the top 10 most frequent head and tail entities in the two KGs. As Figures 9 and 10 show, The top 10 head entities are mostly countries for both KGs. COVID-19 is also an entity that appears in the Top 10 for both KGs. In terms of top 10 tail entities, we see Amharic KG has "human" as the most frequent entity, indicating the KG mostly includes information about individuals followed by "year" indicating there are a lot of head entities that are years. The Tigrinya KG top 10 tail entities are mostly comprised of country names, indicating the KG is mostly represents information about relationships between different countries.

D Zero-Shot Prompts

In this section, we give an example of the prompt we used for the zero-shot experiment in §4.2. For both Aya and GPT-4o, we used the same prompt. For a given target language, we take the reformulated question, Q, and design our prompt as follows:

> Please provide an answer for the following [LANGUAGE] question. Please keep your response to three words maximum and output the answer ONLY. Question: [Q] ?
> Answer:

We attempt to constrain the model output to the tail entity only by instructing the model to output

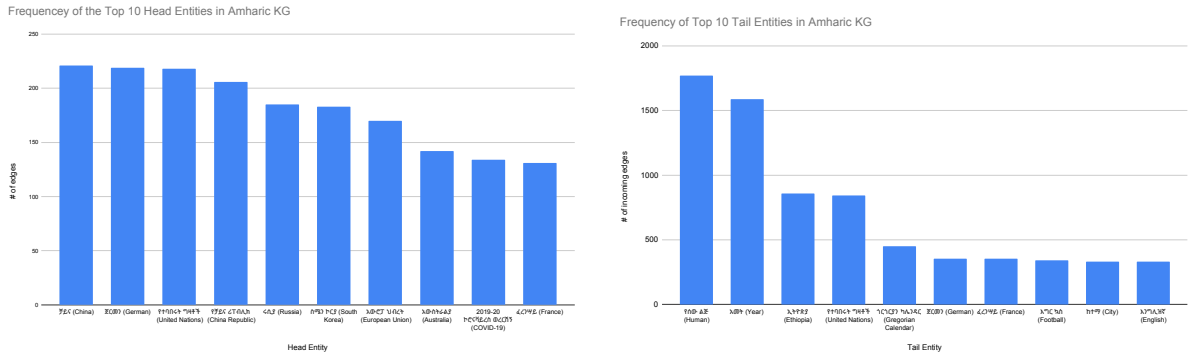

Figure 9: Figure showing top 10 head and tail entities in Amharic KG. English translations for entities are provided in parentheses.

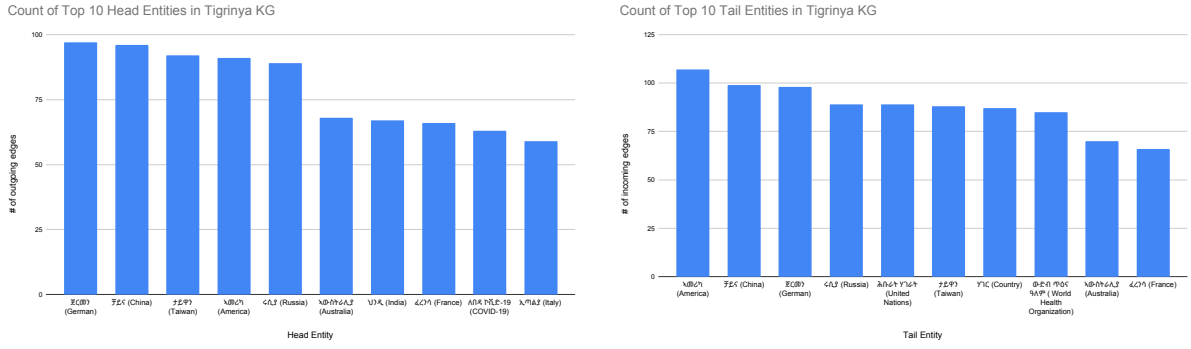

Figure 10: Figure showing top 10 head and tail entities in Tigrinya KG. English translations for entities are provided in parentheses.

Type	%	Examples
Country	77.05	ቤኒን, ኤርትራ, ፓኪስታን, ናሚቢያ, ኤስዋቲኒ
City or Region	13.12	ጀርጂያ, ማድሪድ, ካራካስ, ቲራና, ቦጎታ
Misc.	9.84	ሃይማኖት, ክርስትና, ዓረብኛ, ፖለቲካ, ጉግል

Figure 11: Tail entities that are spelled the same in Amharic and Tigrinya, allowing for shared representation in the model. We find that most of the shared tail entities are names of countries, cities and regions.

	Tigrinya		Amharic	
	% con.	% tail	% con.	% tail
KGT5-Description	49.77	1.78	6.3	0.71
KGT5-One-Hop	48.83	0.89	25.77	1.65

Table 9: Percentage of context extracted from Wikidata Description and through One-Hop connections along with percentage of how many of the contexts have the tail entity. (Con. refers to context.)

the answer only and asking it to keep the prediction at a maximum of 3 words. With manual inspection, we observed that the model might output additional tokens or words. Hence, we adjusted our evaluation function to count a prediction as correct if it contains the tail entity.

D.1 Details on KGT5 Setting

We compared our approach with the scheme used in KGT5 (Saxena et al., 2022a) and KGT5-context (Kochsiek et al., 2023). When comparing the Description and One-Hop connection-based schemes for providing context, we found that the performance did not improve or the two target languages. We hypothesized this could be due to the fact that the knowledge graphs are too small for the models to learn good enough representations on their own and the requirement for structured and labeled data constrained how useful (Kochsiek et al., 2023) approach would be for these low-resourced languages. Table 9 corroborates this hypothesis.