CHALMERS
UNIVERSITY OF TECHNOLOGY

UNIVERSITY OF GOTHENBURG

Generation of Wikidata Descriptions with Grammatical Framework

Master's thesis in Computer science and engineering

Xiao Bokun

Department of Computer Science and Engineering
Chalmers University of Technology
University of Gothenburg
Gothenburg, Sweden 2025

Master's thesis 2025

Generation of Wikidata Descriptions with Grammatical Framework

Xiao Bokun

UNIVERSITY OF
GOTHENBURG

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Computer Science and Engineering
Chalmers University of Technology
University of Gothenburg
Gothenburg, Sweden 2025

Generation of Wikidata Descriptions with Grammatical Framework

Xiao bokun

Supervisor: Inari Listenmaa, Chalmers University of Technology, Computing Science, Computer Science and Engineering
Examiner: Aarne Ranta, University of Gothenburg, Department of Computer Science and Engineering

Master's Thesis 2025
Department of Computer Science and Engineering
Chalmers University of Technology and University of Gothenburg
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Typeset in LaTeX
Gothenburg, Sweden 2025

Generation of Wikidata Descriptions with Grammatical Framework

Xiao Bokun
Department of Computer Science and Engineering
Chalmers University of Technology and University of Gothenburg

Abstract

Wikidata is a collaborative, multilingual knowledge base that serves a wide range of purposes, such as supporting Wikipedia, and plays a significant role in ensuring equitable access to information worldwide. However, the quality and consistency of entity descriptions in different languages vary greatly, and many entities lack descriptions altogether. This thesis presents a workflow based on the Grammatical Framework (GF) for the automated generation of multilingual Wikidata entity descriptions. The system integrates property extraction, grammar design, and automatic linearization, enabling the systematic generation of multilingual descriptions while reducing, to some extent, the need for manual intervention and GF-specific expertise. Manual evaluation shows that, compared to human-written Wikidata descriptions and those generated by large language models, GF-generated descriptions achieve higher cross-linguistic consistency and factual accuracy. The workflow also supports efficient extension to additional languages, as demonstrated by the Bengali case by Mohammad Rakib Imtiaz. These results highlight the potential of GF, the GF Resource Grammar Library, and the approach introduced in this thesis for scalable, verifiable, and reliable multilingual description generation.

Keywords: Wikidata, Grammatical Framework, Natrual Language, computer science.

Acknowledgements

First and foremost, I would like to express my sincere gratitude to my supervisor, Dr. Inari Listenmaa, for her invaluable support throughout this thesis. Her expertise in Grammatical Framework, insightful advice, and patient advice were essential to the completion of this work. While learning about computer science from her, I also hope to learn from her patience and passion for life. I am also grateful to Professor Aarne Ranta for proposing such an interesting and practically significant research topic, as well as for his insightful suggestions and encouragement during this work.

I would also like to thank all collaborators involved in the project, including members of the data group, Xu Huatai and Cao Yuxiang, whose analysis of Wikidata formed a key foundation for this work. My thanks also go to Mohammad Rakib Imtiaz, whose contributions to the Bengali grammar greatly informed my estimates for the scalability of future project expansions.

Finally, I wish to thank my mother for her unwavering support—both emotional and financial—during my studies. "While we were both grieving the loss of my father, she continued to support me, to whom I dedicate this thesis. May my father, her husband rest in peace.

Xiao bokun, Gothenburg, 2025-07-09

Contents

List of Figures

List of Tables

1

Introduction

1.1 Background

The rapid development of the internet and smart devices has fundamentally changed how knowledge is recorded and disseminated. In the past, comprehensive sources such as the Encyclopedia Britannica were authoritative but limited in accessibility and scope. Today, platforms like Wikipedia and Wikidata allow instant access to information across languages and domains.

Taking Wikipedia as an example, users can directly search for articles of interest and conveniently switch between multiple language versions. Currently, the majority of Wikipedia' s content is authored by human contributors, with its accuracy maintained through a series of established policies and review mechanisms [1]. However, it is still common to find discrepancies across language versions of the same article or factual inaccuracies in certain entities. Consequently, Wikipedia is actively working to replace manual discussions and edits with texts automatically generated based on structured databases and explicit rules.

Wikidata plays a crucial role in this transformation, and we will provide a detailed introduction to Wikidata later. In brief, Wikidata is built upon a relational database, systematically storing entities along with their properties and relationships, with properties generally devoid of subjective interpretations. Previous research has already explored utilizing Wikidata together with explicit rules to generate Wikipedia texts. For instance, prior works have employed Grammatical Framework (GF) alongside Wikidata to automatically produce Wikipedia entities on specific topics [2].

Wikidata also employs specific measures to ensure the accuracy of its properties. For example, each property includes references whenever possible. Unlike citations in academic papers or essays, references in Wikidata—due to its structured nature—can achieve complete (100%) coverage. Nevertheless, lot of Wikidata's current content—particularly the descriptions associated with each entity—is manually written, leading to issues such as factual inaccuracies, different descriptions across different languages, and incomplete descriptions in certain languages.

We aim to introduce automated methods to address these problems. It is important to note that although large language models (LLMs) have shown impressive natural language generation capabilities, their tendency toward producing plausible yet inaccurate content ("hallucinations") makes them unsuitable for fact-critical tasks like

Wikidata description generation. For this reason, our work adopts GF, a rule-based formalism, to generate multilingual entity descriptions grounded solely in verifiable facts.

1.2 Purpose and Goals

Currently, Wikidata item descriptions are manually added separately for each language, which often results in factual inconsistencies between languages. For example, the English description for Gothenburg is "second-largest city in Sweden and the capital of the Västra Götaland County" , while the Swedish description reads "tätort i Göteborgs kommun, Sverige" ("urban area in the municipality of Gothenburg, Sweden"), which means that they have different meanings. Such discrepancies conflict with Wikidata' s goal of providing a unified and consistent knowledge base. Another issue is that due to differences in the number of speakers of different languages and varying internet usage habits across cultures, the coverage of descriptions in Wikidata is highly limited across languages.

Our goal is to unify the descriptions of Wikidata entities across different languages, fill in missing descriptions, and ensure that all generated content is grounded in reliable facts rather than derived from machine learning or statistical models. To achieve this, we choose GF as our primary tool because it is a rule-based language for Natural Language Generation, enabling predictable content production. Furthermore, GF provides existing resources such as the Resource Grammar Library (RGL) [3], which can be used for this task. As a functional language, GF allows us to define functions for expressing lexical items and grammatical rules, and to compose and linearize them to produce natural language output. We will provide a more detailed introduction to GF and its associated tools in the following sections.

In addition, a higher-level objective of this work is to reduce reliance on manual work. Therefore, we aim to develop a robust and reusable set of code and workflows that will enable future contributors—who are willing to support the community—to work more efficiently and accurately, while also allowing for the verification of generated content. To this end, we strive to design the grammar in terms of broad, generalizable categories. Since the grammar is organized by topics such as human or places, our ideal workflow requires only a small number of concrete grammar editors who have a basic understanding of GF's concrete syntax for the languages they know; a slightly larger group of contributors who are proficient in a single natural language to handle lexical implementation and review; and only a limited number of contributors familiar with GF, which requires a significant learning effort as a programming language, will be responsible for maintaining the abstract syntax of specific domains. The aiming is to combine the factual accuracy of rule-based generation with improved scalability and efficiency.

Another important challenge is balancing two potentially conflicting goals: reducing reliance on human-written descriptive content and preserving as much raw information as possible from Wikidata, while still producing descriptions that are readable.

We start by analyzing data from Wikidata and identifying the grammatical struc-

tures needed for each topic. Since our team is most familiar with Chinese and English, we first develop the GF concrete syntax for these languages. At the same time, data from Wikidata is analyzed and processed for the grammar design, with entity properties being handled, concatenated, and linearized through Python scripting. The quality of the generated language is assessed by comparing it with existing Wikidata descriptions and outputs from large language models. To support future contributors, we also prepare a comprehensive documentation guide.

2

Theory

This chapter presents the theoretical and technical foundations underlying our approach to multilingual description generation from Wikidata. We first introduce Wikidata as a structured, multilingual knowledge base, highlighting its data access methods via SPARQL queries and JSON APIs. Next, we introduce the Grammatical Framework (GF), focusing on its separation of abstract and concrete syntax and its suitability for multilingual natural language text generation. We also explain the Universal Dependencies (UD) framework, which provides a consistent approach to syntactic annotation across languages. The chapter then describes the GF Resource Grammar Library (RGL), a globally collaborative project contributed to by linguists and developers around the world, which offers a rich collection of syntactic and morphological functions for over 40 languages. Together, these components form a robust foundation for building and evaluating multilingual text generation systems, enabling both semantic accuracy and broad language coverage.

2.1 Wikidata

Wikidata is a collection of structured data–comprising over 100 million entries including data entries, properties, and statements [4]. These data are available in about 287 languages. Entries such as 'Gothenburg' and 'Sweden' are connected by facts. In this way, a description for 'Gothenburg' can be 'second-largest city in Sweden and capital of the Västra Götaland County' in English, as stated in Wikidata [5]. Wikidata entries and facts are increasingly used as the foundation for Wikipedia articles, allowing content to be generated from data rather than written manually. This approach is expected to improve the quality of Wikipedia, particularly its consistency across different languages.

Data can be accessed by searching for its unique identifier [4]. In Wikidata, each item (also known as an "entity") represents a real-world concept or object and is assigned a unique identifier (such as Q25287 for Gothenburg). Every item includes labels in multiple languages (for example, Göteborg' in Swedish and Gotemburgo' in Spanish), along with a brief description in each language. Each Wikidata item contains a series of statements. A statement is a fact about the item, composed of a property (the type of information, e.g., "population" or "part of") and a value (which can be a string, a date, a number, or a link to another item). For example, the statement using the property part of (P361) for Gothenburg (Q25287)

points to the item Västergötland (Q213551), indicating that Gothenburg is part of Västergötland. Some other statements are descriptive—for instance, the property population (P1082) provides the population of Gothenburg, often accompanied by references or qualifiers.

RDF is a standard framework for representing information as a graph of interconnected data points, where data is stored as *triples*—each consisting of a subject, predicate, and object. This model enables RDF databases to handle irregular, flexible, and highly interconnected data structures, making it ideal for knowledge graphs like Wikidata. For example, in Wikidata, entities can be related through complex, hierarchical relationships—such as an item being an instance of a subclass, which itself may be a subclass of another class, and so on. SPARQL, the query language for RDF-based data, allows users to retrieve information that spans multiple layers of such relationships. For instance, a SPARQL query can retrieve all items that are instances of a given class or of any of its subclasses, regardless of how many levels deep the hierarchy goes [6]. This recursive querying capability is crucial for exploring the rich and sometimes deeply nested structures in Wikidata (see Figure 2.1). Another reason is that, as a global-scale project, Wikidata holds over 100 million data entries, which are inherently irregular and challenging to model with traditional relational databases [7].

Below is an example (Figure2.1, Figure2.2) that demonstrates how to retrieve all universities in Sweden along with their descriptions in different languages together with part of the query result. In this case, the subject is universities (Q3918), the predicate is country (P17) and, the object is Sweden (Q34).

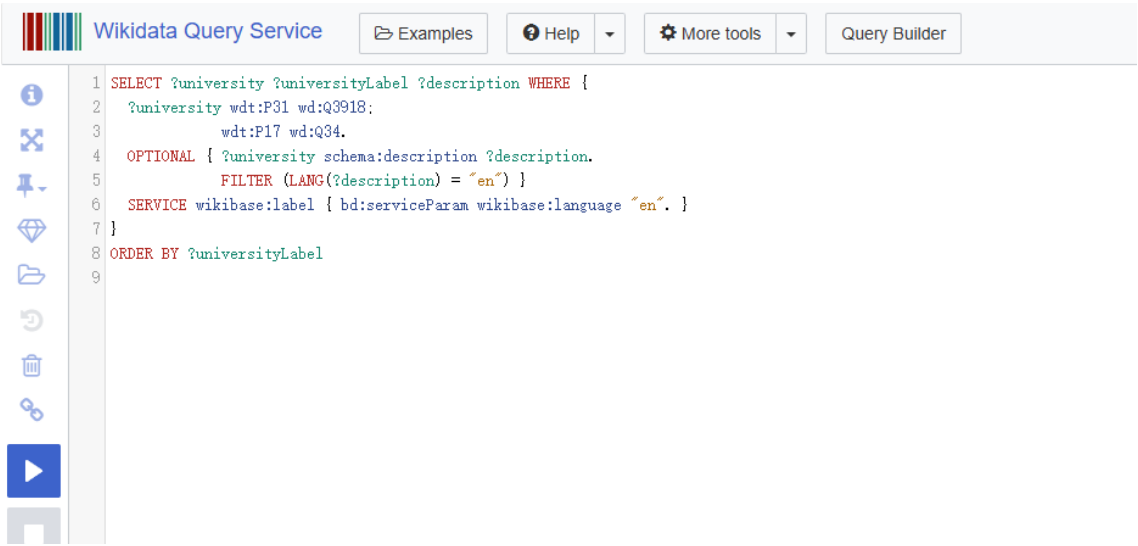

Figure 2.1: SPARQL Commands for selecting Swedish university with labels and descriptions

university	universityLabel	description
wd:Q116681043	Q116681043	institute in Sweden
wd:Q836805	Chalmers University of Technology	university in Gothenburg, Sweden
wd:Q5971267	IIIEE	university institute in Lund University
wd:Q219564	Karolinska Institutet	medical university located in Stockholm, Sweden
wd:Q1782926	Konstfack	art college in Stockholm, Sweden
wd:Q782600	Linköping University	Swedish university
wd:Q1972721	Luleå University of Technology	Swedish university
wd:Q218506	Lund University	university located in the city of Lund in the province of Scania, Sweden
wd:Q977781	Malmö University	university located in Malmö, Sweden
wd:Q1940803	Mid Sweden University	Swedish university
wd:Q854280	Royal Institute of Technology	university in Stockholm, Sweden
wd:Q221645	Stockholm University	state university of Stockholm, Sweden
wd:Q1312762	Swedish University of Agricultural Sciences	university in Uppsala, Sweden
wd:Q1144565	Umeå University	public university in Umeå, Sweden
wd:Q371522	University of Gothenburg	university in Gothenburg, Sweden
wd:Q185246	Uppsala University	research university in Uppsala, Sweden
wd:Q3279424	Växjö University	
wd:Q2002420	Örebro University	state university in Örebro, Sweden

Figure 2.2: Result of Above Commands

Wikidata also provides a JSON-based query method, allowing users to directly request data from the URL corresponding to an entity. The response is returned in JSON format and includes all available information about the entries across different languages. In our actual work, we primarily relied on this method.

2.2 Grammatical Framework(GF)

GF [8] is a multilingual grammar development tool designed to define abstract and concrete syntaxes, enabling natural language parsing, generation, and translation. GF is especially suitable for controlled, high-quality multilingual text generation due to its formal, type-driven approach. In particular, GF guarantees an injective mapping from abstract syntax trees (ASTs) to natural language expressions: as long as the AST is factually correct, the generated natural language will also be factually correct. Moreover, since the same AST can be linearized into multiple languages, this ensures that descriptions in different languages are semantically equivalent and consistent, which is critical for applications requiring cross-lingual reliability and factual alignment. In the GF, the key components include:

- **Abstract Syntax**: The abstract syntax defines the semantics of words as types and specifies the semantics of more complex expressions using functions.

```
-- a 'Hello World' grammar
abstract Hello = {
```

```
  flags startcat = Greeting ;

  cat Greeting ; Recipient ;

  fun
    Hello : Recipient -> Greeting ;
    World, Mum, Friends : Recipient ;
}
```

- **Concrete Syntax**: A concrete syntax corresponds to the lexicon of a specific language, describing how sentences are constructed and mapping specific semantics to corresponding words.

```
concrete HelloEng of Hello = {

  lincat Greeting, Recipient = {s : Str} ;

  lin

    Hello recip = {s = 'hello' ++ recip.s};
    World = {s = 'world'} ;
    Mum = {s = 'mum'} ;
    Friends = {s = 'friends'} ;
}
```

The above GF code can parse the string 'hello world' into the AST `Hello World` and can also generate 'hello world' in other languages using the defined corresponding concrete syntaxes.

Like other functional programming languages, GF features type and category system. In the example below, `cat` is used to define categories, while function declarations are made using types composed from these categories. GF also supports table syntax and record types, which define structured objects with named fields.

In natural language, a lexical item may appear as a single surface word, but often changes form depending on syntactic context. In GF, we define a word as an object of a basic type, where the type is determined by its part of speech and grammatical features. For example, in Chinese:

```
woman_N = mkN " 女人" " 个";
```

This makes use of overloaded functions:

```
mkN = overload {
  mkN : (man : Str) -> N
    = \n -> lin N (regNoun n ge_s) ;
  mkN : (man : Str) -> Str -> N
    = \n,c -> lin N (regNoun n c)
};

regNoun : Str -> Str -> Noun = \s,c -> {s = word s ; c = word c};
```

Here, `s` is the surface form of the noun, and `c` is the classifier, similar to the word "cup" in the phrase "a cup of tea" .

For verbs, more complex constructions are used. For example, here are the functions involved in defining buy (type signatures are omitted for clarity).

```
buy_V2 = dirV2 (irregV "buy" "bought" "bought");
prepV2 v p = lin V2 {s = v.s ; p = v.p ; c2 = p.s ; isRefl = v.isRefl};
dirV2 v = prepV2 v noPrep;
```

This defines the verb *buy* as an irregular transitive verb with its full morphological paradigm.

GF also provides *smart paradigms*, which automatically select the appropriate morphological form based on string patterns. This is especially useful for handling irregular inflections in natural language:

```
smartVerb : Str -> Verb = \v -> case v of {
  _ + ("s"|"z"|"x"|"ch")      => s_regVerb v ;
  _ + "ie"                    => ie_regVerb v ;
  _ + "ee"                    => ee_regVerb v ;
  _ + "e"                     => e_regVerb v ;
  _ + ("a"|"e"|"o"|"u") + "y" => regVerb v ;
  _ + "y"                     => y_regVerb v ;
  _                           => regVerb v
};
```

This mechanism enables GF to generate appropriate verb forms without manually specifying all exceptions.

2.3 Universal Dependencies(UD)

UD is a framework designed for the consistent annotation of grammar, encompassing parts of speech, morphological features, and syntactic dependencies, in different human languages. UD treats words as fundamental units of grammatical annotation. Within the UD framework, the basic annotation units are syntactic words, rather than phonological or orthographic words. For example, in French, the word 'au' is a single phonological and orthographic unit. However, in Universal Dependencies, it is split into two syntactic words: 'à' (ADP) and 'le' (DET), where 'à' serves as a preposition, and 'le' serves as a determiner.[9]

UD is based on lexicalism, where words are categorized into nominals (e.g., subjects and objects), clausal elements (e.g., adjectives, adverbs, and prepositional phrases), and other structural dependencies (e.g., coordination, punctuation, and auxiliary verbs). Additionally, UD follows dependency grammar, where each sentence has a root, typically the main verb or predicate. All other words are linked to this root through syntactic relations, forming a dependency tree. Each word has a head, which establishes its grammatical relationship within the sentence structure.

UD defines 40 dependency relations, systematically categorizing these connections.

As a result, UD enables the annotation of words across different languages, providing a standardized foundation for cross-linguistic syntactic and semantic analysis.

Figure 2.3 illustrates a Universal Dependencies (UD) annotation for a French sentence, as presented in [9]. This diagram visually represents the syntactic structure of the sentence, indicating the grammatical relations between words using labeled arrows. Each word is annotated with its part of speech, morphological features (such as gender, number, and tense), and dependency relation to the syntactic head. The root of the sentence is marked, and punctuation and functional words are shown with their respective roles. Such annotation provides a standardized, cross-linguistic framework for analyzing sentence structure, enabling consistent comparison and computational processing across different languages.

Figure 2.4 provides the same sentence in CoNLL-U format, demonstrating how the same information can be represented in a tabular, machine-readable form widely used in computational linguistics.[9].

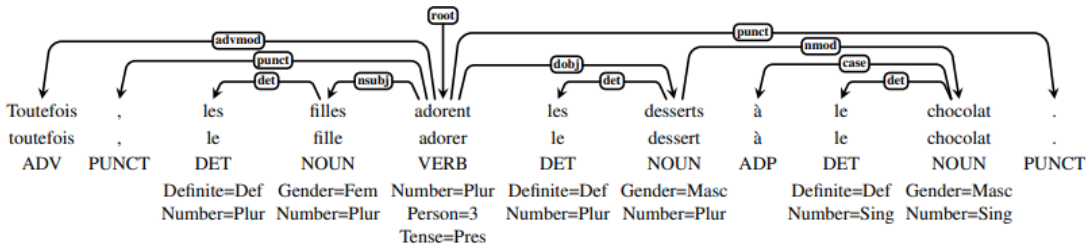

Figure 2.3: UD annotation for a French sentence. (Translation: However, girls love chocolate desserts.)

```
ID  FORM      LEMMA     UPOSTAG XPOSTAG FEATS                                  HEAD DEPREL DEPS MISC
1   Toutefois toutefois ADV     _       _                                      5    advmod _    _
2   ,         ,         PUNCT   _       _                                      5    punct  _    _
3   les       le        DET     _       Definite=Det|Number=Plur               4    det    _    _
4   filles    fille     NOUN    _       Gender=Fem|Number=Plur                 5    nsubj  _    _
5   adorent   adorer    VERB    _       Number=Plur|Person=3|Tense=Pres        0    root   _    _
6   les       le        DET     _       Definite=Def|Number=Plur               7    det    _    _
7   desserts  dessert   NOUN    _       Gender=Masc|Number=Plur                5    dobj   _    _
8-9 au        _         _       _       _                                      _    _      _    _
8   à         à         ADP     _       _                                      10   case   _    _
9   le        le        DET     _       Definite=Def|Gender=Masc|Number=Sing   10   det    _    _
10  chocolat  chocolat  NOUN    _       Gender=Masc|Number=Sing                7    nmod   _    _
11  .         .         PUNCT   _       _                                      5    punct  _    _
```

Figure 2.4: The French sentence from Figure 3 in CoNLL-U format

2.4 GF Resource Grammar Library(RGL)

GF comes with its standard library of morphological and syntactic functions, called the **Resource Grammar Library (RGL)** [3]. RGL is implemented for over 40 languages, ensuring that linguistic structures can be represented in a unified way while allowing for language-specific variations.

The structure and style of RGL are influenced by the Universal Dependencies (UD) framework, classifying words by their syntactic relationships. Based on these relationships, RGL provides functions for composing larger syntactic units.

GF-RGL offers a set of lexical constructors and syntactic functions that cover a wide range of languages. In addition to previously mentioned constructors like `prepV2`, here are some commonly used examples:

```
-- Functions with explanations
mkCN : A -> N -> CN          -- |Combines an adjective (A) and a noun (N)
                                 into a common noun phrase (CN).|
mkA : Str -> A               -- |Converts a string into an adjective.|
mkN : Str -> N               -- |Converts a string into a noun.|
mkCl : NP -> A -> Cl         -- |Forms a clause from a noun phrase and
                                 an adjective predicate.|
mkNP : Det -> N -> NP        -- |Forms a noun phrase from a determiner
                                 and a noun.|
mkRCl : RP -> VP -> RCl      -- |Forms a relative clause from a relative
                                 pronoun and a verb phrase.|

-- Adjective modifying a noun
mkCN (mkA "big") (mkN "house")
-- Output: "big house"

-- Adjective as a predicate
mkCl (mkNP (mkN "water")) (mkA "wet")
-- Output: "Water is wet"

-- Verb taking a complement
mkCl (mkNP theSg (mkN "guest")) want_VV
  (mkVP (mkV "eat") (mkNP theSg (mkN "apple")))
-- Output: "The guest wants to eat an apple."

-- Relative clause
mkNP theSg
  (mkCN (mkN "woman") (mkRCl that_RP (mkVP (mkV "sleep"))))
-- Output: "the woman that sleeps"
```

You can also find definitions for some basic lexical entries in the RGL. For instance, in addition to `buy_V2`, here are other examples of lexical entries of English:

```
father_N2 = mkN2 (mkN masculine (mkN "father")) (mkPrep "of") ;
fear_VS   = mkVS (regV "fear") ;
find_V2   = dirV2 (irregV "find" "found" "found") ;
```

For example, in English, predicative adjectives require the copula *to be*, as in *"water is very wet"*. In contrast, in languages such as Chinese, the copula is often omitted in the present tense, producing a structure like *"water very wet"*. Despite these surface differences, both expressions correspond to the same underlying AST in GF code.

This design enables contributors to define a function in the abstract syntax that takes a noun, an adjective, and an adverb indicating degree, and call the pre-written functions in concrete syntax of different languages. As a result, contributors can generate grammatically natural expressions across different languages without spending time on low-level syntactic rule implementation.

This modular architecture provides several advantages. Developers working on the abstract syntax are not required to manually implement lexical and syntactic details for each individual language. Instead, they can focus on the semantic composition of functions, reusing existing constructs and paradigms provided by the RGL. As a result, contributors with only limited knowledge of GF can still participate effectively by building on top of RGL resources, while more experienced developers can concentrate on extending or refining the RGL itself—improving the core infrastructure that benefits all projects relying on it.

3

Methods

This chapter describes the methods used to construct and evaluate rule-based multilingual entity descriptions from Wikidata. The project involved collaboration with other teams (particularly those led by Cao Yuxiang and Xu Huatai)[10], particularly during data collection, preliminary analysis, and comparative evaluation. Our workflow consisted of: (1) extracting and analyzing relevant entities and their properties from Wikidata, (2) designing abstract and concrete grammars in GF for English and Chinese, and (3) developing automated scripts for property extraction, mapping to grammar functions, and linearization into natural language. Process automation and documentation were further improved to ensure project scalability and maintainability.

3.1 Data Analysis and Selection Criteria

Our methodological foundation is a rigorous data-driven approach to the selection of topics and properties for multilingual entity description generation. With collaborators from the data group, we implemented a multi-stage analysis pipeline to ensure that the chosen categories and attributes would support high-quality, semantically consistent, and cross-linguistically robust descriptions.

The first step in our process involved large-scale data extraction from Wikidata using SPARQL queries and Python scripts, following best practices for efficient and reproducible data collection. To overcome the limitations of single-language or shallow queries—such as incomplete subclass coverage and missing labels—we adopted techniques such SPARQL sentences (e.g., using `wdt:P31/wdt:P279*` for instance-of chains) and iterative pagination to build a comprehensive dataset. All collected data were stored in JSONL format, facilitating structured analysis and later integration with the description generation pipeline.

A key challenge encountered was the frequent absence of labels or descriptions in certain target languages, particularly for entities in underrepresented regions or languages. To address this, data group supplemented missing values using trusted external resources, specifically GeoNames and OpenStreetMap, by implementing entity matching based on labels or external IDs (e.g., GeoNames ID). This enrichment step significantly improved label coverage and data completeness.

For property selection, we first analyzed the frequency of all properties associated

with entities in each topic (such as cities, universities, persons, islands, and lakes). For every topic, all property identifiers (PIDs) were extracted and ranked according to their occurrence across relevant entities. This frequency analysis revealed that a small set of high-frequency properties—typically the top 20—accounts for the majority of available data within each domain. As an illustration, see table 3.1 shows the top 15 most frequent properties among mathematicians in Wikidata, where attributes like "occupation" , "instance of" , "sex or gender" , and "date of birth" dominate the knowledge graph for this category. We intentionally excluded infrequent or overly specific properties to maintain the generalizability and conciseness of generated descriptions.

Table 3.1: Top 15 most frequent properties among mathematicians in Wikidata

Property	**Count**
occupation	17,070
instance of	17,067
sex or gender	17,067
date of birth	16,802
given name	16,105
country of citizenship	15,798
place of birth	15,278
ISNI	14,456
languages spoken, written	13,407
educated at	13,196
maintained by WikiProject	12,665
family name	12,604
date of death	12,233
employer	11,818
place of death	10,298

To further guide description template design, we examined the relationship between structured properties and actual human-written descriptions. More specifically, for the domains of cities and universities, we first extracted the top 20 high-frequency property identifiers (PIDs) from structured Wikidata data (e.g., P17 for "country"). Based on these properties, we constructed an entity description corpus containing the English texts for all relevant entities. For each city or university, we then checked whether the label of each high-frequency property appeared in its corresponding human-written description, matching the property label as a complete word or phrase. The results show substantial variation in coverage across different properties. For example, administrative and geographic attributes such as "country" (P17) were the most frequently observed in descriptions, with 12,096 matches in the city corpus, indicating that administrative affiliation information is nearly ubiquitous in city descriptions. In contrast, other properties exhibited much lower (e.g., " GeoNames ID" , " VIAF ID" , etc.).

Our topic selection was motivated by practical and coverage considerations. We prioritized entity types that are (1) well-represented and structurally consistent in

Wikidata, (2) of broad interest across languages, and (3) suitable for concise, informative description templates. The resulting dataset, enhanced by external enrichment and structured frequency analysis, formed the basis for the subsequent development of multilingual grammatical templates and automated description generation workflows. The details of the property frequency statistics and their domain-specific selection are documented in the Results section.

3.2 GF Grammar Design

In designing the GF grammar, we developed both the abstract grammar and the concrete grammars in English and Chinese for descriptions for specific topics. The abstract syntax serves as an interface: it defines the minimal set of types and functions needed to represent the main informational components for each entity type (such as university, city, or person). This layer is intentionally kept as simple and general as possible, avoiding language-specific distinctions—such as case, grammatical gender, or complex agreement—so as to maximize reusability and extensibility across languages.

In the concrete grammar for each language, the way each category is linearized can be freely defined according to the needs of the target language. This means that the realization of grammatical categories, word order, and inflectional features can be customized for each language, while the overall logical structure specified by the abstract syntax remains consistent.

For example, the abstract syntax for university descriptions may be defined as follows:

```
UniversityDescription :
        UniversityKinds -> Location -> Attribute -> Description ;
-- Used to generate the complete description

university_Kind : UniversityKinds ;
publicKind      : UniversityKinds ;
privateKind     : UniversityKinds ;
-- Usually noun types, representing different categories of universities

CityCountryLocation : City -> Country -> Location ;
CountryLocation         : Country -> Location ;
noLocation              : Location ;
-- These functions generate different location
        adjuncts depending on available information,
-- typically of the Adv type

FoundedIn : Int  -> Attribute ;
-- Takes an integer (the founding year) and
        returns an adjunct describing the founding date,
-- typically of type Adv
```

In the English concrete grammar, the implementation of the above categories and

functions is as follows:

```
Description = CN ;
Location = Adv ;
UniversityKinds = CN ;
Attribute = Adv ;

City = NP ;
Country = LinCountry ;

CityCountryLocation city country = SyntaxEng.mkAdv in_Prep
                                    (joinByComma city country.s) ;
CountryLocation country = SyntaxEng.mkAdv in_Prep country.s ;
noLocation = ParadigmsEng.mkAdv "" ;

university_Kind  = mkCN (mkN "university") ;
publicKind   = mkCN (mkN "public university");
privateKind  = mkCN (mkN "private university") ;

UniversityDescription kind location attr = \
        mkCN (mkCN kind location) attr ;

FoundedIn year = mkAdv (mkPrep "founded in") (symb year) ;
noAttr = ParadigmsEng.mkAdv "";

joinByComma : NP -> NP -> NP  ;
joinByComma = mkNP comma_Conj ;

comma_Conj : Conj = and_Conj ** {s2 = bindComma} ;
```

In the Bengali concrete grammar, the implementation is defined differently; the Bengali grammar was provided by Mohammad Rakib Imtiaz.

```
Description = Utt ;
Location = NP ;
UniversityKinds = CN ;
Attribute = Adv ;

City = NP ;
Country = LinCountry ;

CityCountryLocation city country = mkLocation city country.s ;
ProvinceCountryLocation province country = mkLocation province country.s ;
CountryLocation country = mkLocation country.s ;
noLocation = R.emptyNP ;

university_ben = mkCN (mkN "বিশ্ববিদ্যালয়");

publicKind = mkCN (mkN "পাবলিক বিশ্ববিদ্যালয়");
```

```
privateKind = mkCN (mkN "প্রাইভেট বিশ্ববিদ্যালয়");

UniversityDescription kind location attr =
    cnUtt (mkCN (NounBen.PossNP kind location) attr) ;

FoundedIn year = mkAdv founded_in_year_Prep (symb year) ;
noAttr = ParadigmsBen.mkAdv "" ;

mkLocation = overload {
    mkLocation : NP -> NP -> NP = ApposNP ;
            -- country.NOM , city.(CASE open)
    mkLocation : NP -> NP = id NP
} ;
```

As code above, once the information retrieved from the web is assembled into the expression

```
UniversityDescription university_Kind (CountryLocation Q34_Sweden_Country)
(FoundedIn "1829")
```

And the visualized tree is as follows

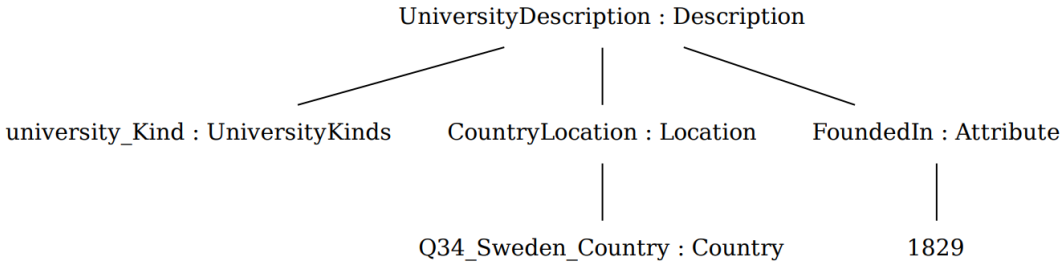

Figure 3.1: Visualized Tree

Which can be linearized as *university in Sweden founded in 1829*.

Another point worth noting is that, although inflectional changes of noun gender are rare in English and Chinese, such variations are common in other languages, such as German or French. Therefore, our grammar also defines gender parameters, which can be implemented as follows:

```
HumanGender = {s : Str ; g : GenderParam} ;
Person = {s : Str ; g : GenderParam} ;
Professions = GenderParam => N ;

# Example implementations
PersonBuilding str gen = {s = str.s ; g = gen.g} ;
Male = {s= ""; g = MaleParam} ;
Female = {s= ""; g = FemaleParam} ;
Unknown = {s= ""; g = UnknownParam} ;
```

```
-- English doesn't have gender distinction in most job titles.
Mathematician = \_ => mkN "mathematician" ;

-- French have gender distinction
Mathematician_FR = table {
  MaleParam => mkN "mathématicien" ;
  FemaleParam => mkN "mathématicienne" ;
  UnknownParam => mkN "mathématicien" ;
};
```

For example, the call

```
Mathematician ! (PersonBuilding "peter" Male).g
```

will generate the noun 'mathématicien' corresponding to the appropriate gender.

When only a subset of nouns in a language are inflected for gender, the linearization category (lincat) can be defined with a gender parameter, which is simply ignored for those nouns whose surface forms do not vary by gender. Besides, for languages that do not encode gender distinctions at all, the abstract syntax must still include the HumanGender parameter, so that the abstract syntax tree (AST) is the same for all languages. But in the concrete grammar for such languages, the lincat of HumanGender is an empty record, and all functions that take a HumanGender as argument would ignore the argument completely. So that way, the abstract syntax functions Male/Female/Unknown are present in the trees, but ignored in the linearizations.

In this way, the gender parameter is present in the trees but does not affect the concrete output for languages without gender distinctions.

3.3 Automated Data Processing and Generation

The GF compiler compiles grammar files into `.pgf` and `.gfo` files, which can be read by Python scripts via the `pgf` package. This allows us to compose GF function calls as string expressions within Python, evaluate them using the PGF interface, and linearize the output into natural language descriptions. Our approach centers on the use of GF's abstract and concrete grammars, not only to define the syntactic structure of descriptions but also to modularize lexical items as functions.

In order to generate concrete vocabulary entities and categories such as `City`, `Province`, `CityKind`, and `Country`, we write Python scripts that extract structured data directly from Wikidata. For types such as `province` and `country`, this process is relatively straightforward. For example, all first-level administrative divisions can be collected by selecting those whose `subclass of (P279)` property is `first-level administrative division (Q10864048)`. The following SPARQL query can be used for this purpose:

```
SELECT DISTINCT ?region ?regionLabel ?labelZh ?nativeZh ?offNameZh ?type
```

```
?typeLabel WHERE {
  ?type wdt:P279* wd:Q10864048 .
  ?region wdt:P31 ?type .
  OPTIONAL { ?region rdfs:label ?labelZh . \
        FILTER(LANGMATCHES(LANG(?labelZh), "zh")) }
  OPTIONAL { ?region wdt:P1705 ?nativeZh . \
        FILTER(LANGMATCHES(LANG(?nativeZh), "zh")) }
  OPTIONAL { ?region wdt:P1448 ?offNameZh . \
        FILTER(LANGMATCHES(LANG(?offNameZh), "zh")) }
  SERVICE wikibase:label { bd:serviceParam wikibase:language "en". }
}
```

After some basic string processing, the data can be directly written into grammar files. However, generating types such as `CityKind` is more complex. Due to the manual editing of Wikidata, there are thousands of different `instance of` items under all city entities, and each city may belong to several types. For example, in the case of Kyoto:

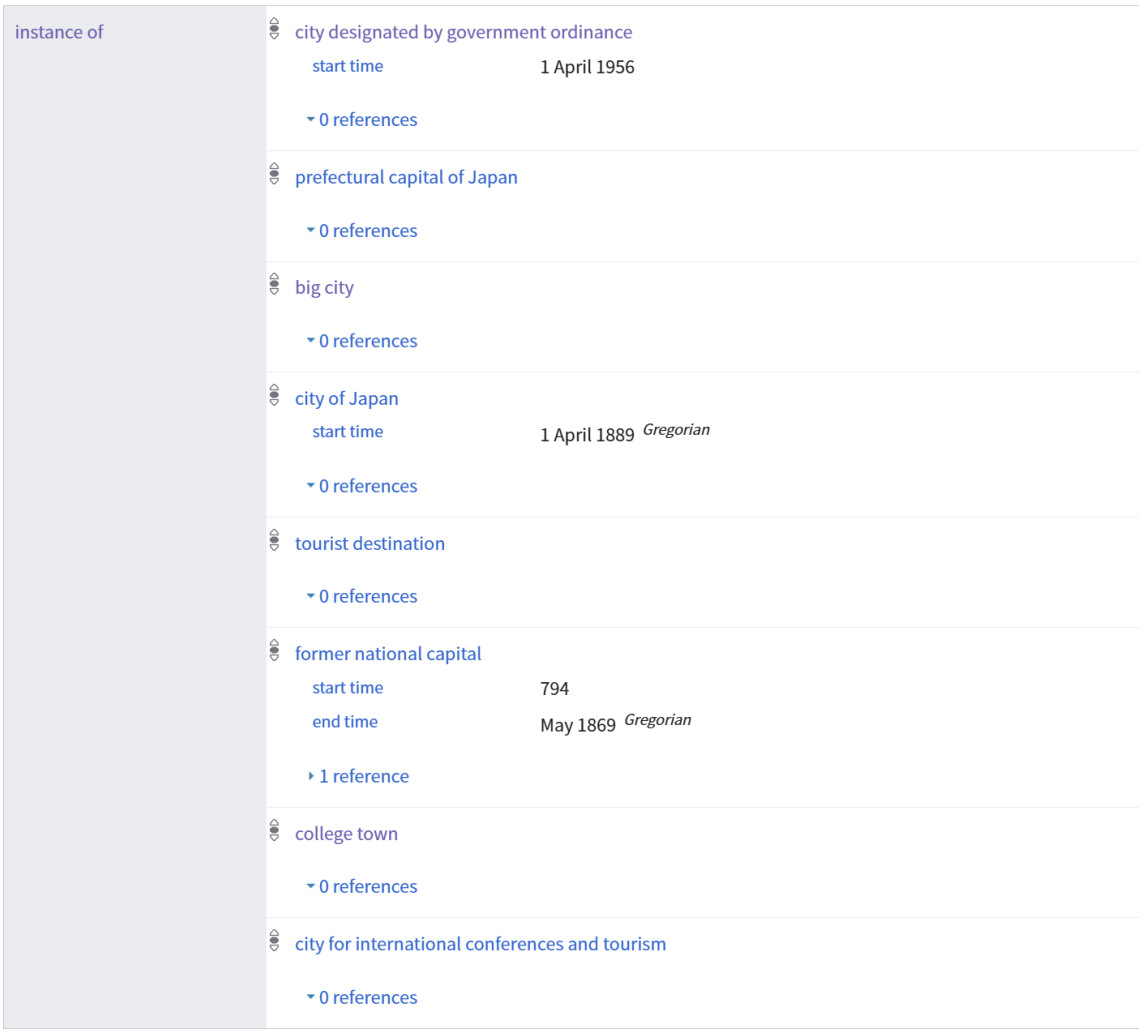

Figure 3.2: The `instance of (P31)` property for Kyoto

If we simply extract the labels or descriptions for these items, the resulting expressions can be very cumbersome and hard to read, such as "city designated by government ordinance in the Kyoto Prefecture, Japan." Currently, there is no fully automated solution for this part. Our current approach is to count the frequency of different `instance of (P31)` items across entities, select those occurring above a certain threshold, and manually translate them for inclusion in the grammar. Fortunately, the workload for this manual step is manageable. In the future, the project may still require some manual intervention, but editors would only need to modify the content inside the brackets for the relevant language file entities (i.e., the translation of specific nouns), which does not require knowledge of GF. For entities such as cities, types of cities, countries and nationalities, provinces, biological families and genera, as well as water bodies, we generated corresponding grammar entities in GF, and translated these grammars to ensure linguistic accuracy.

This functional way, rather than direct string concatenation, is motivated by several factors: First, some properties of items within the same topic can be irregular in Wikidata, which poses challenges for generating descriptions, with, for example, universities classified under districts or other non-standard parent entities. By maintaining a curated table of components, we ensure only entities of clear descriptive value are used. Second, for higher specific items such as smaller towns, the number of editors and the frequency of updates are lower, leading to possible inaccuracies; in contrast, entities like countries or provinces tend to have more reliable data. Moreover, our approach makes human proofreading and post-editing straightforward. Many labels in various languages on Wikidata are either inaccurate or incomplete. Third, numerous labels are not suitable for direct use as lexical items in generated descriptions, as illustrated by the Kyoto example above, directly using instance-of values without post-processing can produce cumbersome and unreadable descriptions. Addressing these issues inevitably requires manual intervention and proofreading. The manually edited results can also be uploaded as improved labels to the relevant Wikidata item lebels. In addition, for languages with grammatical categories such as gender or inflection, it is necessary for contributors to manually curate certain nouns and verbs—such as profession names—which cannot be extracted from Wikidata. It would be unrealistic to attempt to handle these cases through simple string concatenation, considering that there may be thousands of professions worldwide.

The data for generating descriptions is collected using Python scripts that query Wikidata for JSON data on the relevant entities. For each target, the required properties (such as instance type, location, founding year, etc.) are extracted, with further processing when needed. When generating descriptions for human, special attention is needed for nationality and birthplace. In many cases, especially for historical figures, the item's birthplace may belong to a historical country that no longer exists. For example, the mathematician Gauss, born in the Holy Roman Empire, is today commonly described as being from Germany. Thus, we maintain a mapping table to manually associate historical countries with their modern equivalents. Another representative example is that the capital of (P1376) property of Shanghai (Q8686) points to Shanghai International Settlement (Q2170587),

which was in fact a historical area during the colonial period—specifically, means the International Settlement of the British Concession. In reality, Shanghai is now a direct-administered municipality (Q1208802) in China. Once all necessary propoties are collected, the script selects the corresponding functions from the grammar, or simply by passing the original string as an argument. These components are then combined according to the defined grammar rules to produce descriptions in the target language. This approach ensures both accuracy and readability, as all components have been vetted and structured according to the grammatical constraints of the target languages. It also enables flexible extension to new languages or entity types by editing only the relevant grammar files and component lists.

4

Results

This chapter presents a multilingual workflow for generating Wikidata entity descriptions using the Grammatical Framework (GF). By automating property extraction, writng GF grammars and automated property analysis and linearization, the workflow significantly increases coverage of accurate English and Chinese descriptions. Manual evaluation across multiple topics demonstrates that, compared to both existing Wikidata descriptions and LLM-based methods, GF-generated descriptions offer higher factual accuracy and cross-linguistic consistency, though sometimes at the expense of descriptive richness than existing Wikidata descriptions.

4.1 Workflow

The main goal of this project is to ensure equitable access to information across different languages and, under the premise of accuracy, to generate Wikidata descriptions in as many GF-RGL-supported languages as possible. To this end, we have streamlined the workflow as much as possible, improved code reusability, and minimized both the level of GF expertise required and the amount of manual work needed.

As a reference point, Mohammad Rakib Imtiaz—the Bengali RGL contributor who is familiar with both GF and the Bengali language—completed all Bengali concrete grammar for the seven topics in approximately eight person-days based on the existing codebase[11]. Considering that Mohammad Rakib Imtiaz did not fully reuse my scripts, and the fact that the contributor manual had not yet been written at the time he joined, we believe that contributors for other languages who already know GF could complete the work even more quickly.

Table 4.1 provides an overview of each step in the workflow, summarizing the required GF expertise and indicating the extent to which each step relies on Wikidata.

For greater clarity, we include below the pseudocode for two essential steps: the extraction of GF noun functions from Wikidata and the generation of multilingual descriptions.

Task	Contributor Requirements	Relation to Wikidata
Extraction of GF nominal functions from Wikidata, translation and validation	Little or no GF knowledge	Collected from Wikidata, with partial manual processing
Writing abstract GF grammars	Familiarity with GF is required, the actual workload is minimal, as repetitive code does not need to be rewritten	Reference to existing descriptions and property frequency
Writing concrete GF grammars for different languages	Familiarity with GF	Not required
Generation of descriptions	None	Fully based on Wikidata

Table 4.1: Overview of workflow steps, contributor requirements, and Wikidata usage.

Extraction of GF Noun Functions from Wikidata

Algorithm 1 Extraction of GF Noun Functions from Wikidata

Require: SPARQL query for all instances of subclasses of Q10864048 (province-level administrative units)

1: Send SPARQL query to Wikidata endpoint, requesting English and Chinese labels for each region
2: Initialize a map `grouped` to store entities grouped by (type_qid, type_label)
3: **for** each result in the query response **do**
4: Extract region QID and English label
5: Obtain Chinese label using the following priority:
- Use Wikidata label if available
- Else use native or official name if available
- Else, call Google Translate API to translate the English label into Chinese
- If still unavailable, fall back to English label

6: Generate a valid GF function name from QID and English label
7: Add to `grouped` under the corresponding (type_qid, type_label) group
8: **end for**
9: **for** e **do**ach group in `grouped`, write the following files:
- `CityRegions.gf`: abstract syntax listing all functions
- `CityRegionsEng.gf`: English concrete syntax using `mkNP (mkPN "...")`
- `CityRegionsChi.gf`: Chinese concrete syntax using `mkNP (mkPN "...")`

10: **end for**

In addition, we used Wikipedia and the Google Translate API to supplement textual content for some functions lacking entity labels in the target language. Considering that Wikipedia also relies on manual editing and the reliability of Google Translate is limited, we did not upload these automatically generated contents to Wikidata.

Generation of Descriptions

For each topic, we selected the most representative and high-frequency properties to construct the descriptions. Table 4.2 summarizes the set of properties used for each topic.

Topic	**Selected properties**
City	city type, capital status, province-level division, country
University	university type, province-level division, country, year of establishment
Person	profession, gender, place of birth, nationality, birth and death years
Island	body of water, province-level division, country
Lake	province-level division, country
Biological genus	family
Biological species	genus, family

Table 4.2: Selected properties for each topic

Based on these selected properties, the following procedure is used to automatically generate descriptions for each entity. For each topic, the relevant property values are retrieved, mapped into a GF expression according to the predefined template, and linearized into the target languages. The main steps are outlined below.

Algorithm 2 Generation of Descriptions

Require: QID of an entity
Ensure: Multilingual description strings
1: entity ← fetch Wikidata entity by QID
2: entity_type ← detect type using SPARQL
3: **for** properties required by entity_type:
4: Use recursive search, mapping, or qualification checks to obtain each property
5: (e.g., finding the required parent administrative entities, mapping historical countries to modern countries)
6: **end for**
7: Compose a GF expression using processed properties
8: **for** each target language:
9: Linearize the GF expression
10: **end for**
11: **Return** Multilingual descriptions (e.g., English, Chinese)

4.2 Comparison of description results

This section presents a manual evaluation of description results. For the topics of city and university, we compare the original Wikidata descriptions, the LLM-generated descriptions by the teams of Xu Huatai and Cao Yuxiang (using RAG without further finetuning)[10], and the GF-based descriptions. For island, lake, and human topics, we compare only the GF-based descriptions and Wikidata. To make the evaluation more representative, we also included one less prominent entity in each group as a reference. The evaluation focuses on accuracy, consistency, and richness.

- **Accuracy:** Whether the description is factually correct and free of hallucinations.
- **Consistency:** Whether descriptions are coherent and aligned across different languages.
- **Richness:** The degree to which the description provides comprehensive and informative content.

4.2.1 City

Accuracy: The general-purpose LLM using only RAG exhibited severe hallucination issues: in the English descriptions, Beijing and Guangzhou were incorrectly identified as universities; in the Chinese descriptions, Gothenburg was placed in Albania, and factual errors appeared in the population figures for Stockholm. Both the GF-generated descriptions and those from Wikidata were highly accurate. However, the GF-generated Chinese description for Kanazawa is not sufficiently accurate, which is due to the existence of multiple terms for "capital" in Chinese.

Consistency: The Chinese and English descriptions generated by the LLM were inconsistent with each other. In contrast, the GF-generated descriptions were consistent across languages. In Wikidata, the descriptions for Guangzhou and Stockholm are identical in both languages, while the Chinese description for Beijing and Kanazawa, and the English description for Gothenburg provide more detailed information.

Richness: Although the LLM-generated content tended to be more verbose, it contained numerous factual errors. The GF-generated descriptions offered more detailed information than Wikidata for Guangzhou and Gothenburg in Chinese.

Label	LLM_en	GF_en	Wikidata_en	LLM_zh	GF_zh	Wikidata_zh
Beijing	Beijing Huajia University is a university located in China, founded in 1985.	capital of People's Republic of China	capital city of China	北京是中国的首都，人口为 1538 万。	中华人民共和国的首都	中华人民共和国的首都暨直辖市
Guangzhou	The Guangzhou Open University is a university based in China, which was established in the year 1961.	capital of Guangdong, People's Republic of China	capital city of Guangdong Province, China	广州是中国广东省的首都，人口为 1308 万	中华人民共和国广东省的首府	中国广东省省会
Gothenburg	Gothenburg is a city located in Sweden (Wikidata Q6580), known for its rich history and vibrant culture.	capital of Västra Götaland County, Sweden	second-largest city in Sweden and capital of the Västra Götaland County	哥德堡是阿尔巴尼亚科尔察州的一个市镇，人口为约 20,848 人。	瑞典西约塔兰省的首府	瑞典的城市
Stockholm	Stockholm is a city that has been featured in various works of fiction.	capital of Sweden	capital and largest city of Sweden	斯德哥尔摩位于瑞典（Sweden），居民为 350,859 人。	瑞典的首都	瑞典首都
Kanazawa	Kanazawa is a city in Ishikawa Prefecture, Japan	capital of Ishikawa Prefecture, Japan	capital city in Ishikawa Prefecture, Japan	一个在日本金泽㊣的市区，人口约为 37 万	日本石川縣的首府	位於日本石川縣中部的城市，为石川縣的縣廳所在地

Table 4.3: Comparison of city descriptions

4.2.2 University

Accuracy: The LLM also exhibited hallucination problems, with factual errors appearing in the Chinese description of the University of Cambridge and Jiamusi University, both the Chinese and English descriptions of Tsinghua University and Chalmers University of Technology. In addition, the LLM incorrectly included QID information from RAG. Both the GF-generated descriptions and those from Wikidata were accurate. **Consistency:** The Chinese and English descriptions generated by the LLM were inconsistent with each other, while the GF-generated descriptions were consistent across languages. The descriptions in Wikidata, however, differed between languages with all entities. **Richness:** Although the LLM-generated descriptions were generally longer, they included many factual errors. In contrast, the GF-generated descriptions offered more comprehensive information in the English description of Tsinghua University, the Chinese descriptions of the University of Cambridge and the Massachusetts Institute of Technology, and in all descriptions of Chalmers University of Technology and Jiamusi University. Furthermore, the Chinese description of Tsinghua University and the English description of the University of Cambridge in Wikidata were more detailed than those generated by GF.

Label	RAG_en	GF_en	Wikidata_en	RAG_zh	GF_zh	Wikidata_zh
Tsinghua University	Tsinghua University is a university located in Beijing, China (Q2541), established in 1911 (P580).	public university in Beijing, People's Republic of China founded in 1911	public university in Beijing, China	该实体是美国休斯顿大学清湖分校（University of Minnesota Twin Cities），位于美国明尼苏达州（Minnesota），成立于 1971 年，属于美国的一所公立大学。	建立于 1911 年的中华人民共和国北京市的公立大学	中华人民共和国北京市综合性大学
University of Cambridge	The University of Cambridge is a higher education institution located in England (Q878), more specifically, in the city of Cambridge (Q4091) with no recorded year of inception (inception: null).	public university in United Kingdom founded in 1209	collegiate public research university in Cambridge, England, United Kingdom	剑桥大学位于英国，成立于 1502 年	建立于 1209 年的英国的公立大学	1209 年创建的一所学校
Chalmers University of Technology	Chalmers University of Technology is a Swedish technical university located at http://www.wikidata.org/entity/Q3403 (Göteborg, Sweden), founded in 1967.	university in Sweden founded in 1829	university in Gothenburg, Sweden	查尔姆斯理工大学位于奥地利（Q40），在维也纳市内（Q131266）。该大学于 1995 年成立。	建立于 1829 年的瑞典的大学	瑞典私立大学
Massachusetts Institute of Technology	The Massachusetts Institute of Technology (MIT) is a prestigious higher education institution based in Cambridge, United States, founded in 1861.	private university in Cambridge, United States founded in 1861	private university in Cambridge, Massachusetts, USA, founded 1861	Massachusetts Institute of Technology is an educational institution located in the United States, with the Wikidata entity Q143.	建立于 1861 年的美国剑桥的私立大学	1861 年建立的一所学校
Jiamusi University	Jiamusi University is a university located in Heilongjiang, China, founded in 1950	university in Jiamusi, People's Republic of China founded in 1947	university in China	这是一个来自伊朗的高等教育机构，位于维基数据中的实体 Q3918	建立于 1947 年的中华人民共和国佳木斯市的大学	中国黑龙江省佳木斯市普通高等学校

Table 4.4: Comparison of university descriptions

4.2.3 Island

Accuracy: Both the GF-generated descriptions and those from Wikidata are accurate.

Consistency: The GF-generated descriptions are consistent across languages, while the Wikidata descriptions differ between languages. Moreover, the Chinese descriptions for Greenland and Yuzhny Island are missing from Wikidata.

Richness: The English description of Greenland in Wikidata is more detailed. For Honshu, both the English and Chinese descriptions in Wikidata are richer. For Madagascar and Yuzhny Island, the GF-generated descriptions in both English and Chinese provide more information.

Label	GF_en	Wikidata_en	GF_zh	Wikidata_zh
Greenland	island in Kingdom of Denmark	island located between the Arctic Ocean and the North Atlantic Ocean	丹麦王国的岛屿	
Honshu	island in Japan	largest island of Japan	日本的岛屿	日本最大島及本土四島之一
Madagascar	island in Madagascar, located in Indian Ocean	island in the Indian Ocean	马达加斯加的岛屿，位于印度洋	马达加斯加岛
Yuzhny Island	island in Arkhangelsk Oblast, Russia, located in Arctic Ocean	island in Russia	俄罗斯阿尔汉格尔斯克州的岛屿，位于北冰洋	

Table 4.5: Comparison of island descriptions

4.2.4 Lake

Accuracy: The GF-generated descriptions for oth Lake Superior and Lake Victoria in English and Chinese contain errors, as Lake Superior and Lake Victoria are located on borders and cannot be clearly attributed to a single country; the remaining descriptions are accurate. All Wikidata descriptions are accurate.

Consistency: The GF-generated descriptions are consistent across languages, while the Wikidata descriptions differ between languages except for Lake Tele.

Richness: Excluding factual errors, the descriptions from Wikidata are more detailed.

Label	GF_en	Wikidata_en	GF_zh	Wikidata_zh
Lake Superior	lake in Ontario, Canada	largest of the Great Lakes of North America	加拿大安大略省的湖泊	加拿大湖泊
Lake Baikal	lake in Buryatia, Russia	freshwater rift lake in southern Siberia, Russia	俄罗斯布里亞特共和國的湖泊	俄罗斯湖泊
Lake Victoria	lake in Tanzania	lake in east-central Africa	坦桑尼亚的湖泊	非洲淡水湖
Lake Tele	lake in Republic of the Congo	lake in Republic of the Congo	刚果共和国的湖泊	刚果共和国湖泊

Table 4.6: Comparison of lake descriptions

4.2.5 Human

Accuracy: Both the GF-generated descriptions and those from Wikidata are accurate.

Consistency: The GF-generated descriptions are consistent across languages, while all the Chinese and English descriptions in Wikidata differ.

Richness: The GF-generated descriptions include birth and death dates as well as place of birth in more cases. However, GF-generated descriptions lack important features that are difficult to extract from properties but are significant—such as "creator of the theory of relativity" or "first President of South Africa."

Label	GF_en	Wikidata_en	GF_zh	Wikidata_zh
Albert Einstein	American male theoretical physicist, philosopher of science and inventor born in Germany (1879 –1955)	German-born theoretical physicist (1879–1955)	美国的男理论物理学家、科学哲学家、发明家 (1879 –1955)，出生于德国	猶太裔美國物理學家，相對論創立者
Ada Lovelace	British female mathematician, programmer and poet (1815 – 1852)	English mathematician (1815–1852)	British Female mathematician, programmer and poet born in Roman Empire (1815 –1852) 英国的女數學家、程序员、诗人 (1815 –1852)	英国数学家
Nelson Mandela	South African male politician (1918 –2013)	First President of South Africa and anti-apartheid activist (1918–2013)	南非的男政治人物 (1918 –2013)	第 1 任南非總統、反種族隔離革命家
Shiing-Shen Chern	American male mathematician born in People's Republic of China (1911 – 2004)	Chinese-American mathematician and poet (1911-2004)	美国的男數學家 (1911 –2004)，出生于中华人民共和国	數學家（1911 年至 2004 年）

Table 4.7: Comparison of human descriptions

4.3 Description Coverage by Topic

We provide statistics on the supplementation of entities in Wikidata that originally lacked Chinese or English descriptions, as shown in Table **??**. The columns represent the five topics (city, university, island, lake, and human), and each cell shows the supplemented number, total number, and coverage rate in the format (supplemented / total / rate). For the "human" topic, due to the large number of entities, descriptions have currently only been generated for those whose profession includes "mathematician." The results for Bengali are derived from Mohammad Rakib Imtiaz [11].

Topic	**English**	**Chinese**	**Bengali**
City	5291 / 32475 / 16.3%	26484 / 32475 / 81.6%	25824 / 32496 / 79.5%
University	351 / 14184 / 2.5%	9689 / 14184 / 68.3%	13202 / 14184 / 93.0%
Island	4579 / 50000 / 9.2%	49179 / 50000 / 98.4%	6267 / 10001 / 62.7%
Lake	1339 / 50000 / 2.7%	12447 / 50000 / 24.9%	5438 / 10001 / 54.4%
Human	6900 / 42327 / 16.3%	39126 / 42327 / 92.4%	/ / /

Table 4.8: Coverage of supplemented descriptions (supplemented / total / rate) for each topic and language

5

Conclusion

5.1 Discussion

This thesis presents a study on the automated generation of multilingual Wikidata descriptions, leveraging Wikidata's structured data as the foundation and utilizing the GF to produce accurate and consistent, descriptions. The workflow included three main steps—property extraction, grammar design, and automated linearization—and the results were evaluated, leading to several findings.

Strengths: The evaluation results indicate that Wikidata descriptions in different languages may differ from one another, and LLM-generated descriptions inevitably tend to exhibit hallucinations, occasionally introducing errors or misleading content. In contrast, descriptions generated by GF maintain cross-linguistic consistency and, for many topics, provide more information while remaining readable; moreover, GF is extensible. On the other hand, this work has reduced the GF expertise required for future contributors—those unfamiliar with GF can still participate in reviewing and verifying nominal functions. In addition, the work of Bengali RGL contributor Mohammad Rakib Imtiaz demonstrates that individuals with GF experience can fast develop the concrete grammar for a language they are familiar with and generate hundreds of thousands of descriptions. With ongoing improvements to the scripts and the addition of contributor manuals, future expansions to additional languages will proceed even more efficiently.

Weaknesses: However, several limitations were also founded. Firstly, factual errors still occur in GF-generated descriptions. For example, lakes located on borders may be incorrectly described as belonging to a particular country, or certain important features that cannot be extracted from properties—such as the fact that Mandela was the first president of South Africa—are not reflected in the descriptions. Secondly, a certain amount of manual work is still required, including manual proofreading of the automatically generated grammar and manual judgment in the final generation scripts, rather than relying purely on data. The introduction of more manually defined rules can improve quality, but also introduces ambiguity or edge cases. Thirdly, due to the ongoing updates to Wikidata, the generated descriptions may lag behind the most recent edits, affecting timeliness. Fourthly, the current approach does not fully utilize the high-quality descriptions already available in Wikidata: some manually written descriptions could be retained or enhanced, while those of lower quality should be replaced. However, the system currently lacks

an effective mechanism for such conditional judgment. Finally, the current scripts require multiple consecutive API requests and are subject to file size limitations, resulting in suboptimal efficiency for large-scale generation.

5.2 Conclusion

In summary, this thesis presents a GF-based workflow for the automated generation of multilingual entity descriptions in Wikidata and conducts a manual evaluation. Empirical results show that this approach outperforms manual editing (in terms of consistency and coverage) and large language models (in terms of factual accuracy) across various entity types and languages.

Looking ahead, several directions can further enhance this workflow. First, selectively analyzing and leveraging existing human-written descriptions—retaining high-quality entities while automatically generating or replacing those that are incomplete or less reliable—holds significant promise. Second, it would be beneficial to fully separate manual intervention from the core automated pipeline, allowing for independent and parallel improvements to rules and data. Third, technical advances —such as parallelization, fully automated scripts, and the integration of external knowledge bases, along with mechanisms to ensure information reliability (e.g., attribution of responsibility)—could improve both efficiency and reliability. Together, these measures would contribute to a fully automated, scalable, and verifiable multilingual description generation pipeline.

Bibliography

[1] Wikipedia contributors, *Wikipedia: Policies and guidelines*, Accessed: 2025-05-17, 2025.

[2] K. Angelov, A. Carrión del Fresno, E. Voloshina, and A. Ranta, "Leveraging grammatical framework and wordnet for natural language generation from wikidata," in *International Symposium on Distributed Computing and Artificial Intelligence*, Springer, 2024, pp. 173–184.

[3] A. Ranta, "The GF Resource Grammar Library," *Linguistics in Language Technology*, vol. 2, 2009. [Online]. Available: `http://elanguage.net/journals/index.php/lilt/article/viewFile/214/158`.

[4] D. Vrandečić and M. Krötzsch, "Wikidata: A free collaborative knowledgebase," *Commun. ACM*, vol. 57, no. 10, pp. 78–85, Sep. 2014, ISSN: 0001-0782. DOI: `10.1145/2629489`. [Online]. Available: `https://dl.acm.org/doi/pdf/10.1145/2629489`.

[5] Wikidata, *Gothenburg*, [Online]. Available: urlhttps://www.wikidata.org/wiki/Q25287, Dec. 2024.

[6] W. contributors, *Wikidata:sparql query service - wikidata*, Dec. 2024.

[7] W3C, *Resource description framework (rdf): Concepts and abstract syntax*, Available at: urlhttps://www.w3.org/TR/rdf-concepts/, 2014.

[8] A. Ranta, "Grammatical Framework," *Journal of Functional Programming*, vol. 14, no. 2, pp. 145–189, 2004.

[9] J. Nivre, M.-C. de Marneffe, F. Ginter, *et al.*, "Universal Dependencies v1: A multilingual treebank collection," in *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, Portorož, Slovenia: European Language Resources Association (ELRA), May 2016, pp. 1659–1666. [Online]. Available: `https://aclanthology.org/L16-1262/`.

[10] C. Y. Xu Huatai, "Analysis and generation of wikidata descrip- tions," 2025.

[11] M. R. Imtiaz, "Wikidata descriptions and bangla resource grammar," 2025.