Leveraging Grammatical Framework and WordNet for Natural Language Generation from Wikidata

Krasimir Angelov[1,2] krasimir@chalmers.se,
Andrea Carrión del Fresno[1,2] guscarrian@student.gu.se,
Ekaterina Voloshina[1,2] ekaterina.voloshina@chalmers.se, and
Aarne Ranta[1,2] aarne.ranta@chalmers.se

[1] University of Gothenburg
[2] Chalmers University of Technology

Abstract. We present a prototype for the generation of Wikipedia articles from Wikidata in multiple languages. For the generation we leverage the multilingual resources in the GF Resource Grammars Library (RGL), the GF WordNet and the existing labels in Wikidata. The result is a system where we get an out-of-the-box baseline generation for currently 26 languages which can be quickly adjusted by hand to produce publishing quality documents. We evaluate six of the languages.

1 Introduction

Wikipedia is one of the most visited sites on the Web and provides more than 50 million articles in about 300 different languages. The coverage of the different language editions is, however, unevenly distributed. For example, the top nine languages comprise more than half of all articles, while the bottom half of all languages accounts for less than 10%. The number of articles does not fully represent the scale of the problem. While for many topics there are extensive articles in English, for other less developed language editions, the corresponding articles are just stubs. Articles in smaller Wikipedia editions are also more likely to be out of date [14].

The opposite happens as well – small places, local celebrities and customs are best described in the local language and may not appear at all in English. In 2020, Wikimedia has announced the Abstract Wikipedia project which aims to improve the situation by making it possible to generate articles from structured data.

We present a prototype[3][4] and a methodology where articles for countries are generated from information in Wikidata. The method is thus limited to topics which are well represented in Wikidata. When that is not the case, it is possible in principle to let the authors edit an abstract language independent representation which is then rendered separately for each language. That would correspond to the notion of constructors in [14] but we leave that as future work.

The prototype relies on the linguistic information in the Resource Grammars Library (RGL) in the Grammatical Framework (GF) [11,12], the GF WordNet [3,2], and the entity labels in Wikidata.

[3] https://cloud.grammaticalframework.org/wikidata/index.wsgi
[4] https://github.com/krangelov/gf-wikidata

2 Related Work

An obvious remedy to the language problem is to use machine translation, and in fact Wikipedia has its own translation tool[5]. The use of the tool, however, has been highly contested since the output from any automatic translation is not reliable and can be used only after post-editing. Unfortunately people forget that and publish unedited content. The tool also does not solve the problem that when the world changes, articles must be updated, and we still need human editors for that even if they do use automatic translation. Generation with Large Language Models faces the same quality problem [15] and therefore the approaches in Abstract Wikipedia aim at more controllable methods.

In order to better explore the design space, the Abstract Wikipedia encourages the development of alternative approaches. One option is CoSMo [4] – a proposal for a graphical language in the spirit of UML for content selection. Similar to CoSMo, we do consider an extension of the GF language for content selection, but for now we simply use Python and we leave the language as a future work.

Ninai/Udiron [9] is another system which uses tree editing for content selection, while trying to reuse as must as possible the information available in Wikidata. The tree editing is a technique also available in GF and has been used for a long time [6]. More recently, however, it has been much more popular to use parsing for content creation [1]. People simply find it easier to work with text rather than syntactic or semantic trees. In our vision, content creation should be a combination of light-weight programming in addition to parsing text examples.

A major difference between CoSMo, Ninai/Udiron and our approach is that while the former start from scratch, we reuse and extend an existing system which gives us the advantage of all syntactic and lexical resources created in the past three decades. An early work based on GF [13] showed how these can be used. Here we extend that by integrating more tightly with Wikidata which gives us the best of both worlds.

The main effort in Abstract Wikipedia is now dedicated to Wikifunctions[6] – a wiki of code which allows functions to be edited and executed online. This is also the place where all the language generators will be hosted in the future.

3 Language Resources

3.1 The Resource Grammars Library and GF-WordNet

The GF Resource Grammars Library (RGL) [11,12] is a community project being developed for years, and currently provides the syntax and the morphology for about 40 languages. Unfortunately in terms of lexicon it only includes a small test set with a couple of hundred words.

The lexicon for the project comes from the GF WordNet [3,2] and contains about 110 000 expressions mostly coming from the Princeton WordNet [5]. We preserve the same semantic relations, but while the original WordNet is composed of lemmas, in GF WordNet the basic unit is the abstract function.

[5] https://en.wikipedia.org/wiki/Wikipedia:Content_translation_tool
[6] https://www.wikifunctions.org

Similar to the concept of a synset, the abstract function represents a particular meaning generalizable across languages. In each language the function is then mapped to an inflection table with a structure compatible with the RGL. Unlike the synset, which can include words synonymous only in a particular context, the abstract functions link words which are translations of each other in as wide a context as possible. In that sense, the abstract function is a concept tighter than a synset and is useful for translation and multilingual generation. Abstract functions with a similar meaning are still grouped in a synset like in Princeton WordNet.

The lexicon currently contains 28 languages, but all languages except for English are automatically generated from already existing translation dictionaries and WordNets with the help of automatic disambiguation [3]. Mistakes are thus possible and as part of the current project we also did corrections in the lexicon. Swedish and Bulgarian are notable exceptions which have been extensively checked beforehand [2], and currently more than 50% of the entries are already checked. Another exception is Finnish which was created from the sense-aware translations in FinnWordNet.

3.2 Integration with Wikidata

During the GF WordNet development, many of the noun synsets were aligned with Wikipedia articles, which helps in disambiguation. When we started working with Wikidata, we merged our alignment with Wikidata's own alignment. In the process many mistakes were identified and fixed on both sides. As result, currently Wikidata has 30 377 links to Princeton WordNet 3.1. Other lexemes that exist in GF WordNet but not in Princeton WordNet were linked directly with Wikidata entities. We use the resulting links for lexical selection during the generation.

Another missing piece is that while WordNet contains names of people and places, those are limited to important historical figures and well-known places. Although names often propagate with little change from one language to another, there are still differences. This is quite common for place names, but it also happens for people names when two languages use different scripts. We filled in the gap by importing names of places and personal and family names from Wikidata. When a name is missing in one language, we try to borrow it from another related language. For example, a name for Spanish can be borrowed from French or vice versa. Although not ideal, this is better than providing no verbalization at all. This extension increased the size of our lexicon to 4.3 million entries. See Table 1 for details.

WordNet	adjectives, nouns, verbs, etc.	110 thousand	
Wikidata	**Given names**	64 thousand	Describing
	Family names	531 thousand	7.3 million people
	Place names	3.7 million	
Total		4.3 million	

Table 1: Lexicon size after extending the grammar with Wikidata names

4 System Architecture

GF [12] is a programming language specialized in the description of natural languages. A key concept is the abstract syntax – a language independent representation, which is rendered to one or more concrete languages by using grammars.

The abstract syntax is used on two different levels. The Resource Grammars Library applies it only on the level of linguistic concepts (noun phrases, verb phrases, etc). This means that the library alone provides syntactic translations, which are often reasonable, but they can also fail if the proper translation requires different constructions.

In limited domains, we can go further and build a specialized abstract syntax which is more semantic than syntactic, and can thus hide deeper syntactic differences. On that higher level, the RGL syntax is still useful but only as a library which can abstract away low-level syntactic details. This still allows us to implement most of the semantic abstraction for several languages at once. At the same time when the languages need different constructions, specific parts can be implemented specifically for every language while preserving the common semantics.

The concept of abstract syntax makes the framework attractive for Wikipedia, but we need to face the challenges. To start with, Wikipedia contains a huge diversity of domains – from geographic places and people to math, physics, biology, etc. We expect each new domain to bring its own concepts but we also expect a lot of overlap.

The Wikipedia is also highly dynamic – new content is introduced every minute. Soon or later the new content brings in new language constructions which also need to be supported in the natural language generation. GF on the contrary, like most programming languages, is designed to be statically compiled. This means that a change in the grammar requires a recompilation and a restart of the system.

Lastly, the framework has never been used on the scale of Wikipedia. Typical GF applications involve at most thousands of concepts, while here we have millions.

A related issue is the way we use the RGL as a library in the domain-specific applications. Again like in other programming languages, the library is statically linked with each application. If we have a lot of domains like in Wikipedia, this means a lot of applications where each application contains a version of the library.

In the course of the project, we implemented a new version of the GF compiler and interpreter. While previously, the applications were executed from statically compiled binaries loaded in memory, now an application image is stored in a database file which can be both queried and updated by a specialized engine. This gives us both the scalability and the flexibility needed for the Wikipedia project.

Instead of having a different grammar for each domain, as in [13], we now made it possible to use the library directly from Python. The Python program drives the content selection, and calls into the library for the actual language generation. Whenever different languages require different constructions, the program simply generates different abstract syntax. We still benefit from the fact that the RGL abstracts away most of the differences and from not having to compile separate applications for each domain.

The architecture for the current prototype is shown on Fig. 1 and a screen shot of an article is shown on Fig. 2. From the web page the user can query Wikidata for different entities by name. When an entity is selected, the Python program fetches the entity data, potentially together with other related entities. Based on the data, the program plans

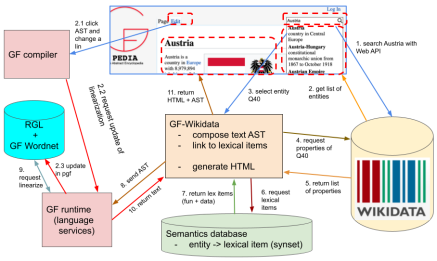

Fig. 1: System Architecture

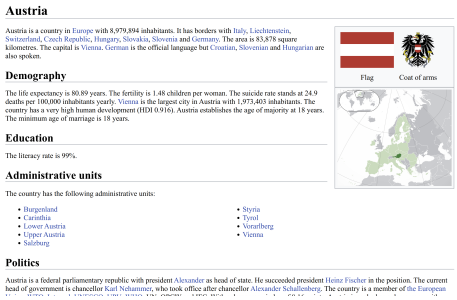

Fig. 2: Screenshot

the document structure and finally produces a sequence of abstract syntax trees. In the process two databases are consulted. The semantic database contains links between the GF WordNet lexicon and the entities in Wikidata. This lets us do the lexical choice when referring to different entities. The second database contains the compiled GF WordNet and the RGL.

The user can also do limited content editing directly from the web page. The "Edit" tab on the web page provides an interface where it is possible to click on words and edit their translations and inflection tables. When this happens, a request is sent to the GF compiler which runs on the server. It processes the new definition of the word and stores the compiled version in the database. Changing the natural language generation itself is possible by updating the Python program. We do not allow that from the web yet, since running arbitrary Python code on the server is a risk. The situation will change when we integrate with WikiFunctions which runs Python code in a sandbox.

5 English Baseline

As a first use case we chose to generate articles about countries in English. Thanks to the multilingual nature of our lexicon and grammar, this also provides an out of the box translation for all other languages but as we will show in the next section, adaptations in each language were also necessary.

Wikidata defines 216 countries, and this small set allows us to do extensive testing while the rich information for each of them lets us to generate longer articles. The content is organized into the following sections:

General information: It distinguishes whether a state is a country or an island state and identifies its location in a specific region of the world, such as Scandinavia, the Middle East, Central Asia, Southern Africa, etc. When there is no more specific region we just use the continent. Additionally, this section also covers information regarding the state's borders with neighboring countries, the total area in square kilometers, the capital city and the official and minority languages spoken in the state.

Demography: This section presents quantitative data regarding life expectancy, fertility, suicide rates, and the Human Development Index (HDI). It also includes other

demographic details such as the capital city and its population, legal ages of majority and marriage, retirement age, and the official religion of the country[7].

Education: We describe the country's education system, beginning with the age range for compulsory school attendance. The literacy rate is also cited, together with statistics on the number of children who are out of the education system.

Administrative units: This is a simple list of the geographical areas into which the state is divided. The list is useful for its hyperlinks to the articles describing each area.

Politics: Here we include information pertinent to the country's form of government and its current and former head of state and government. In addition, the organizations which the country is currently a member of and/or has belonged to in the past are also mentioned. Further data comprises the democracy index, the degree of civil liberties and political rights as stated by the Freedom in the World report, the designation of a terrorist state, when applicable.

Economy: This section helps to understand the country's economic landscape. We can find economic indicators such as the gross domestic product (GDP), Gini coefficient, inflation rate, total reserve, median income and unemployment rate. The official currency is also introduced, along with the value-added tax (VAT) and the income tax.

Climate: The maximum and/or minimum temperatures recorded in the country.

All that information can also be found in a tabular format in Wikidata, but the textual representation has at least two advantages:

- Related facts are grouped in sections or even in a single sentence. The textual representation is therefore more structured, compact and easy to comprehend.
- Whenever possible we also add an interpretation. For instance, parameters like the Gini coefficient, the Democracy Index and the HDI are numbers but they also have categorical interpretations which can be turned into textual descriptions.

6 Language Adaptations

As we move from the English baseline to different languages, we encounter new linguistic issues. We focused on improving the generation for Bulgarian, French, Russian, Spanish and Swedish.

The first and most obvious issue is the need to handle the translation of multi-word expressions – fixed expressions that cannot be split into separate words without altering the meaning or the grammatical structure of the sentence [7]. This characteristic makes the translation task particularly complex, as they often do not translate compositionally. That is, a direct translation often fails to capture the intended concept accurately in its original language. An example is the compound *"life expectancy"* and its Spanish counterpart *"esperanza de vida"*, which literally translates to *"hope of life"*.

The expression 'square kilometres' on Table 2 is another example. In Swedish it is clearly a compound noun – 'kvadratkilometer' , while in English it is ambiguous since WordNet describes 'square' as both an adjective and a noun. In French and Spanish the default translation of a compound noun involves the preposition de – 'kilomètres de

[7] "Official religion" refers to a religion that is formally recognized by the state. An official religion may not be present in all countries.

BUL	Площта е 9 984 670 квадратни километра.
ENG	The area is 9,984,670 square kilometres.
FRE	La superficie est de 9,984,670 kilomètres carrés.
RUS	Площадь – 9 984 670 квадратных километров.
SPA	La superficie es de 9,984,670 kilómetros cuadrados.
SWE	Ytan är 9984670 kvadratkilometer.

Table 2: An illustrative example of linguistic differences between languages.

carré' / 'kilómetros de cuadrado' which is clearly not what we need. On the other hand, using an adjectival modification renders the correct expressions as in the table. The Bulgarian and Russian grammars are more flexible and can translate compound nouns by either replacing the modifying noun with an adjective or by inserting a preposition similar to French and Spanish. This means that regardless of whether the expression is treated as a compound noun or an adjectival modification on the abstract level, we always get the right construction in the concrete language.

Table 2 shows one more difference. When expressing a characteristic or an attribute associated with a particular subject in Spanish and French, we often see the preposition 'de' ('of') following the verb 'es'/'est' ('to be'). On the other hand, the preposition is omitted in the other languages, and the copular verb is directly followed by the attribute.

Generally, prepositions do not translate directly, as they may carry different meanings. For instance, when describing the geographical location of Canada, one might say *"Canada is a country in North America"*. The French translation, however, is *"le Canada est un pays d'Amérique du Nord"*, where the appropriate preposition 'de' (contracted as d') is more similar to the English 'of'.

At the syntactic level, when we want the focus of a sentence to fall on the action over the subject, the passive voice is often used in English. However, there are other grammatical constructions that are preferred in other languages and that are used to express the same idea. That is the case of the pronoun 'se' in Spanish, utilized with transitive verbs in the active voice to indicate that the interpretation of the verb is passive and that the subject is the recipient of the action. This use of 'se' is a form of the passive voice in Spanish, distinct from the traditional passive voice in English ('to be' + past participle). Similar was encountered in Bulgarian and Russian: in some constructions reflexive forms of verbs are used instead of passive voice forms.

Normally these differences in GF are handled in application specific grammars, since covering all differences in the RGL once and for all is too difficult. Since we decided not to use an application grammar, sometimes we simply have to generate different RGL representations from the Python program for each language.

When the differences are more localized we can instead choose to add new multiword expressions in the lexicon. We used that for "life expectancy" for instance. On the other hand, constructing expressions by combining existing words is a more lightweight approach if we can still get the right wording in all languages.

In total we added 10 new expressions in the lexicon, and handled 12 other constructions by altering the generation code in Python.

7 Evaluation

One of the things that we want to evaluate is to what extent the GF WordNet lexicon is appropriate as a baseline for generating multilingual content. We always start the generation with one language – English in this case. Thanks to the language independent API of the RGL this automatically produces an out-of-the-box draft for all the 28 available languages in GF WordNet. The first step for every language is then to go through the articles and check whether we use the correct lexemes. As it turns out most of the words are used correctly but it may happen that they still have uncertain status in the lexicon since they were never checked before. In those cases we just confirm that the word choice for a given sense is correct. If the word choice is wrong or missing then we simply update the lexicon. In either case the validation status or the potentially new word choice is saved in a database and preserved for future applications.

After all words are validated, we get an improved draft, which is almost fluent but some problems still remain because one and the same linguistic construction can be used in different ways in different languages. Finally, we fix non-compositional constructions which results in a final publishing-quality article.

To evaluate the effect of each step, we compute the BLEU scores [10] for each of the drafts against the final article (Table 3). Traditionally, the BLEU score is used to evaluate a machine translation against a human translated document. Here we use it in reverse, i.e. we work on the language generator until we get the final article and then we use that instead of the gold standard. This means that for the final article we always have a score of 100 for all languages which would be very hard in machine translation.

	Initial Draft				Improved Draft				Google Translate			
	1-gram	2-gram	3-gram	4-gram	1-gram	2-gram	3-gram	4-gram	1-gram	2-gram	3-gram	4-gram
Bulgarian	80.04	72.94	67.05	**61.12**	99.17	98.88	98.65	**98.41**	41.53	34.38	30.01	**26.31**
English	94.08	93.13	92.19	**91.19**	100.00	100.00	100.00	**100.00**	-	-	-	**-**
French	64.43	53.94	44.80	**38.01**	95.60	93.45	91.10	**88.75**	76.17	69.32	63.98	**59.99**
Russian	43.21	26.01	17.28	**11.77**	93.12	88.82	85.68	**83.28**	63.75	51.60	42.63	**36.07**
Spanish	73.57	62.48	52.82	**44.75**	96.13	93.75	91.10	**88.31**	82.62	72.99	64.77	**58.26**
Swedish	81.82	76.44	71.67	**67.01**	99.26	98.94	98.75	**98.57**	67.07	56.51	49.23	**43.76**

Table 3: BLEU scores for the generated articles after each phase.

What we observe is that the BLEU scores for the initial draft against the final version varies from 38 to 67 with two exceptions. English is our source language and there we only fixed the capitalizations of some words. This explains the high-scores. The other exception is Russian, where the grammar was recently rewritten from scratch, and during the project we found both incorrect lexemes as well as mistakes in the syntactic rules. As a result the initial scores are much lower than for the other languages.

These scores basically measure how good is the current RGL in combination with the GF WordNet with no additional work. When BLEU is used for machine translation, scores above 50 generally reflect good and fluent translation [8]. Indeed, even without any improvements we found the articles to be quite understandable but still far from production quality.

After fixing wrong word translations or filling in missing ones, we get the improved draft, and there the scores already go above 80. For translation, BLEU scores above 80 are very difficult to obtain because a single sentence can be translated in many possible ways. Since our generation method is predictable, we can control how our output should look like. In that sense it is not surprising that we can get scores up to 100. It is interesting that the effect of non-compositional translations is only 2-17 BLEU points. The difference is of course language dependent but it shows that we can get quite good translations almost for free for many languages.

For comparison we also translated the final versions of the English articles with Google Translate to all the other languages and analyzed the results. As expected the BLEU scores are much lower, but comparing only the numbers would be unfair since Google Translate may also choose to translate in other ways which are also correct. Below, we only highlight cases where the translation is clearly wrong.

Occasionally, it appears that Google's translation of a term may not be appropriate for the context of the sentence, resulting in the incorrect sense of the word. When translating the sentence "the fertility is X children per woman", the term "fertility" is understood as the ratio of live births in an area[8]. In Google's Spanish translation we find the term "fertilidad" ('fertility'), which pertains to the state of being fertile or capable of producing offspring[8]. Hence, this interpretation does not align with the intended meaning in the sentence. The same is also mirrored in the translations of the other languages.

Another notable difference is the inaccuracies or omissions that can occur when translating information from the original text. For example in a sentence describing the official and unofficial languages spoken in a country. Google's translation omits the unofficial languages by stating that many other languages are spoken in the country, and only highlighting the official ones. The result is a translation that does not convey the same meaning as the original text. It appears that long sentences may contribute to the occurrence of hallucinations, as this behavior is not seen in similar but shorter sentences.

In addition, Google's version fails to provide a consistent translation of some acronyms or organization/institution names that have their equivalence in other languages. An example includes the IFC (International Finance Corporation) or the Commonwealth of Nations, which are presented as is, without providing their Spanish equivalents, 'CFI' and 'Mancomunidad de Naciones' respectively.

Finally, several proper names are not capitalized in the translated version. Even though this matter may not directly concern the translation task, it does impact the overall quality of the output text.

8 Conclusion

We find the quality of the generated articles satisfactory and the generation process is easy to steer in the desired direction. Since we have control over the process there is no risk of hallucinations unlike with Large Language Models.

[8] Definitions taken from the Princeton WordNet.

The lexicon is largely automatically generated and thus there are mistakes, but the same is true for any machine learning method. On the other hand, we make it possible to manually correct the lexicon, which means that it will improve over time.

In terms of performance, the bottleneck is in the communication with Wikidata. Every request takes about 0.5 seconds and we need to do several requests. Because of that, the user cannot see the requested page immediately. We know that the Wikifunctions team has plans for additional caching which might improve the situation.

Using Python in combination with GF works, but a special purpose programming language might be better. Currently the Python code has to do a lot of pattern matching to find which combination of properties exists for the current entity and how to package them in compact sentences. This can be done more ergonomically in a specialized language.

References

1. Angelov, K.: Incremental parsing with parallel multiple context-free grammars. In: European Chapter of the Association for Computational Linguistics (2009)
2. Angelov, K.: A parallel WordNet for English, Swedish and Bulgarian. In: Proceedings of the 12th Language Resources and Evaluation Conference. pp. 3008–3015. European Language Resources Association, Marseille, France (May 2020), https://www.aclweb.org/anthology/2020.lrec-1.368
3. Angelov, K., Lobanov, G.: Predicting translation equivalents in linked wordnets. In: The 26th International Conference on Computational Linguistics (COLING 2016). p. 26 (2016)
4. Arrieta, K., Fillottrani, P., Keet, M.: CoSMo: A multilingual modular language for content selection modelling. In: Symposium On Applied Computing (2024)
5. Fellbaum, C.: WordNet: An Electronic Lexical Database. MIT Press (1998)
6. Hallgren, T., Ranta, A.: An extensible proof text editor. In: Parigot, M., Voronkov, A. (eds.) Logic for Programming and Automated Reasoning. pp. 70–84. Springer Berlin Heidelberg, Berlin, Heidelberg (2000)
7. Huddleston, R., Pullum, G.K.: The Cambridge Grammar of the English Language. Cambridge University Press (2002)
8. Lavie, A.: Evaluating the output of machine translation systems. In: Proceedings of Machine Translation Summit XIII: Tutorial Abstracts. Xiamen, China (Sep 19 2011), https://aclanthology.org/2011.mtsummit-tutorials.3
9. Morshed, M.: Using Wikidata lexemes and items to generate text from abstract representations. Semantic Web Journal **3564-4778** (2023)
10. Papineni, K., Roukos, S., Ward, T., Zhu, W.J.: BLEU: a method for automatic evaluation of machine translation. In: ACL. pp. 311–318 (2002)
11. Ranta, A.: The GF resource grammar library. Linguistic Issues in Language Technology (2009)
12. Ranta, A.: Grammatical Framework: Programming with Multilingual Grammars. CSLI Publications, Stanford (2011), iSBN-10: 1-57586-626-9 (Paper), 1-57586-627-7 (Cloth)
13. Ranta, A.: Multilingual Text Generation for Abstract Wikipedia in Grammatical Framework: Prospects and Challenges, pp. 125–149. Springer International Publishing, Cham (2023)
14. Vrandečić, D.: Architecture for a multilingual Wikipedia (2020)
15. Zhang, Y., Li, Y., Cui, L., Cai, D., Liu, L., Fu, T., Huang, X., Zhao, E., Zhang, Y., Chen, Y., Wang, L., Luu, A.T., Bi, W., Shi, F., Shi, S.: Siren's song in the AI ocean: A survey on hallucination in large language models (2023)