From Structured Textual Data to Semantic Linked-data for Georgian Verbal Knowledge

Archil Elizbarashvili[1], Mireille Ducassé[2], Manana Khachidze[1], Magda Tsintsadze[1]

[1]Tbilisi State University, Georgia
[2] Univ Rennes, INSA Rennes, CNRS, IRISA, France
E-mail: archil.elizbarashvili@tsu.ge, mireille.ducasse@irisa.fr, manana.khachidze@tsu.ge, magda.tsintsadze@tsu.ge

Abstract

The Georgian language has a difficult verbal system. To help foreigners learn Georgian, a linked-data base of inflected forms of Georgian verbs is being built: KartuVerbs. We use structured textual knowledge developed by Meurer (2007) that has a much broader scope than KartuVerbs. However, accessing its lexicographic data is challenging; the work on its base has stopped; all properties are not systematically present for every verb; some properties, important for us, do not exist. After filtering and reconstructing some properties, KartuVerbs currently contains more than 5 million inflected forms related to more than 16 000 verbs; there are more than 80 million links in the base. Response times are acceptable when running on a private machine, thus validating the feasibility of the linked-data approach. There is still a need to validate, correct and expand data. Considering the mass of data, this requires tools. This paper presents a process to transform textual structured knowledge into semantic linked data, applied to Georgian verbal knowledge. The process successively applies improvement tools. A specific one, using decision tree technique, complement occasional missing values. The scripts produced so far are freely available. They can be adapted to other applications to help transform data produced for given objectives into other data suited for different objectives.

Keywords: Data transformation; Data validation; Machine learning; Decision tree; Georgian language

1. Introduction

The Georgian language has a difficult grammar. The verbal system, in particular, is challenging. As discussed in more detail in Ducassé & Elizbarashvili (2022), there are numerous irregular verbs. Conjugation can modify both the beginning and the ending of verbs. For example, verb "to work" (mushaoba - მუშაობა), at the first person plural of present tense gives "vmushaobt" (ვმუშაობთ). Note the preradical "v" at the beginning of the verb to mark the first person, and the ending t" to mark the plural. Some tenses, such as future, often introduce a preverb. For example, for verb "to work", the first person singular future is "vimushaveb" (ვიმუშავებ). An "i" has been inserted after the "v" marker of first person. Note that stem formant "ob" has changed into "eb". See for example Tuite (1998) for an exhaustive description. To help foreigners learn Georgian conjugation, a linked-data base of inflected forms of Georgian verbs is being built: KartuVerbs. It is accessible by a logical information system, Sparklis, see Ferré (2017). Sparklis uses

linked-data and enables powerful access and navigation as demonstrated in Ducassé (2020) and Ducassé & Elizbarashvili (2022).

To build KartuVerbs, we started from a structured textual form of the knowledge developed by Meurer (2007) for the Georgian language within the INESS project, called the Clarino base in the following. INESS is an infrastructure to help linguists explore syntax and semantics. It is multilingual and it has a much broader scope than KartuVerbs. However, accessing its lexicographic data is challenging for our target users who are not necessarily linguists. Furthermore, the work on its base for Georgian has stopped. Integrating its data into KartuVerbs both revives them and allow them to evolve. There are more than 60 possible properties, sometimes attached to inflected forms, sometimes attached to verb paradigms. Some of them are obsolete, kept for historical reasons. There are missing pieces of information. All properties are not systematically present for every verb. Some properties, important for us, do not explicitly exist, for example the ending of a form. The initial data were based on the dictionary of Tschenkéli (1965). The Georgian language has evolved since then.

After filtering and reconstructing some properties, KartuVerbs currently contains more than 5 million inflected forms related to more than 16 000 verbs for 11 tenses; each form can have 14 properties; there are more than 80 million links in the base. Response times are acceptable when running on a private machine, thus validating the feasibility of the semantic linked-data approach. There is still a need to validate, correct and expand data. Considering the mass of data, this requires tools. We are currently building experiments using machine learning algorithms.

Section 2 analyses the Clarino database with respect to our needs and introduces a typology of fields. Section 3 describes the transformation process to go from the structured text to the linked data. The process is in 3 blocks. The first block scraps the web pages into a CSV file. The second block aims at incrementally improving the data. The third block produces RDF data and integrates them into Sparklis. Section 4 describes how the decision tree algorithm can help improve a field that has occasional missing values. The field is the verbal noun, the lemma to represent a Georgian verb; there is no infinitive is Georgian. Verbal noun is crucial for our knowledge base. Section 5 discusses further work and Section 6 concludes the paper.

The main contribution of the described work is that all the scripts of the process are freely available on the web[1]. They can be adapted to other applications. Those of the first block could be the base to scrap other textual sources for other languages or applications, not necessarily KartuVerbs. Those of the third block could be used to integrate into KartuVerbs (or another linked-data application) CSV data from other sources than INESS. The scripts to implement the decision tree algorithm dedicated to missing values for verbal nouns could be customized to predict occasional missing values of other fields. Furthermore, the typology of fields described in Section 2 can be used as an analysis grid to help transform data produced for given objectives into another set of data suited for different objectives.

Clarino

...	aorist	vn	...
...	1sg ვაადამიანე, გავაადამიანე	*ადამიანება	...
...	...	...	...

Kartuverbs (CSV)

form	tense	person	number	masdar	...
ვაადამი-ანე	aorist	1	sg	გაადამიანება	...
გავაადამიანე	aorist	1	sg	გაადამიანება	...
...	...	...	...	...	...

Table 1: First singular aorist tense form of verb გაადამიანება (gaadamianeba): Clarino's display and Kartuverbs records

2. The Initial Clarino Base

As already mentioned, the Clarino base is aiming at linguists whereas KartuVerbs is aiming at foreigners learning the Georgian language. Sometimes beginners would have a hard time to interpret Clarino information. For example, we already introduced verbal nouns, the lemmas representatives of verbs. They in general contain a preverb that is important to understand the meaning. As illustrated by Table 1, in Clarino the verbal noun field does not explicitly mention the preverb, because linguists can easily infer the full values of verbal noun with preverbs of the forms. In KartuVerbs, however, we need the full verbal noun, otherwise users will not be able to find the verbs in a dictionary. In the following, we call "masdar" the full version of verbal noun. In the example, Clarino's verbal noun is *ადამიანება (*adamianeba) whereas the masdar is გაადამიანება (gaadamianeba), for verb "to humanize somebody". Furthermore, the textual information we have access to is displayed in a condensed way. For example, as also illustrated by Table 1, all the possible inflected forms of a given verb at a given tense and at a given person are all listed in one field. The linked-data approach of KartuVerbs base requires that the relations are not factorized. For example, instead of the list of inflected forms, there should be as many records as there are inflected forms. Our process, thus, parses and interprets records.

In the Clarino base, the verbs are indexed by roots, a given root in general corresponds to several verbs, and conversely a verb can have several roots. Therefore, Clarino chooses one of the possible roots of a verb as an index. It is called the common root. Verbs have inflected forms in 11 tenses, 6 persons. Table 2 shows the Clarino fields for a form of verb "to humanize somebody". "გააადამიანებს" (gaaadamianebs) is the inflected form at 3rd person singular future. The verbal noun is "*ადამიანება". One of the 3 fields related to preverb gives "გა" (ga). The root is "ადამიან" (adamian). The stem formant is "ებ" (eb). There is no causative stem formant. The Tchkhenkeli Class is T1. Morphology Type is

[1] https://github.com/aelizbarashvili/KartuVerbs

Form	”გააადამიანებს” (gaaadamianebs)	Causative Stem Formant	„_”
Tense	future	Stem Formant	”ებ” (eb)
Person	3rd	Tchkhenkeli Class	T1
Number	sg	Morphology Type	active
Verbal Noun	”*ადამიანება” (adamianeba)	Verb ID	1
Preverb (3 variants)	”-”, ”გა” (ga), ”-”	Common Root ID	4
Root	”ადამიან” (adamian)		

Table 2: Clarino fields for a form of verb ”to humanize somebody”

active. It is the first verb (Verb ID = 1) of the 4th common root (in the index of Clarino, Common Root ID = 4).

The basic field for us is the inflected form. In addition, we use the following form characteristics: Tense, Person, Number, Verbal noun (that we use to build the Masdar), Preverb (3 variants), Root, Stem Formant, Causative Stem Formant, Tchkhen-keli Class, Morphology Type, verb ID, Common Root ID.

The Clarino fields do not exactly fit our needs. They can be classified as follows. Note also that certain verbs do not have all the forms for all the tenses.

1. Fields that we need, that are systematically present and that seem correct; for example, tense, person, number and some linguistic classification inherited from Tschenkeli's work.
2. Fields that we need, that are systematically present but with specific encoding that need systematic (easy) decoding. The main example is the root of the form that, in Clarino, can contain Latin characters in the middle of the Georgian characters. They are used to signal alternatives. Another characteristic is that some verbs have different roots at different tenses. As the base is indexed by roots, Clarino decided on a common root and attached all the possible roots to the verb and not to the forms; after extraction, forms have all possible roots of the verb, all but one being incorrect. Note that correcting these two features can be done by simple scripts.
3. Missing fields but there is enough information in the Clarino fields to deduce the information. For example, preverbs can be deduced from 3 different Clarino fields. Masdars can be deduced from preverbs and verbal noun.
4. Fields that we need, that are systematically present but with occasional mistakes; for example, verb ID. This field category is typical of any source of data. It is almost impossible to create a large body of data and make no mistake.
5. Fields that we need, that are not systematically present but for which the absence can be normal; for example, preverb and stem formant. Forms at present tense often do not have any preverb. Forms at aorist tense often do not have any stem formant.
6. Fields that we need that should always be present, but that are occasionally absent; for example, verbal noun.

7. Missing fields and there is no information in Clarino to deduce them; for example, English infinitive.
8. Fields that we do not need (yet).

In the remaining of this article, Section 3.2 briefly presents the processing of fields of categories 1 to 3. Section 4 describes how machine learning is used to address fields of category 6.

3. Transformation Process

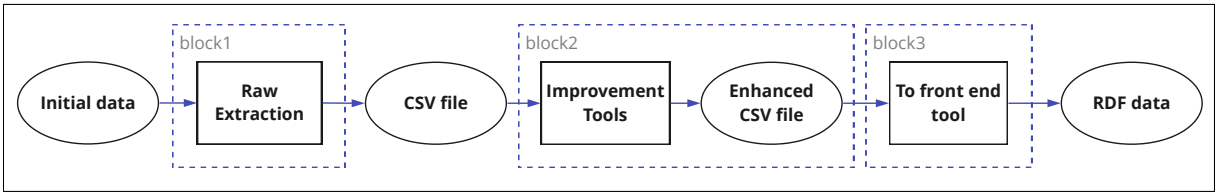

Figure 1: Global structure of the transformation process

As illustrated by Figure 1, the process to transfer data from Clarino to KartuVerbs consists of 3 main blocks. The first one starts from the Clarino web page and generates an intermediate CSV File. The second one consists of several processes to improve the raw data (in relation with issues described in Section 2). The third one transforms the CSV data into RDF data and creates a SPARQL endpoint.

3.1 From Clarino To An Intermediate CSV File

First the Clarino web pages are scraped. The result is a 22 million lines, 625 MB, Json file in pretty format, one line per form with fields. In Clarino, information is hierarchical, whereas our aim is to generate relational data so that information can be accessed from any piece of data (see Ducassé (2020) and Ducassé & Elizbarashvili (2022) for further details). The Clarino structure starts with root, then verb, whereas our key information is inflected form. The process thus flattens the structure and generates tuples whose first field is an inflected form. In Clarino, it is possible to have different values for the same field. In that case, several lines are generated. For example, for a given tense, there may be n possibilities for a given person. In that case, there will be n lines with the same tense and person, and with different forms. Then a Python script converts json to csv. The Clarino properties that are not used for KartuVerbs are filtered out to keep only 14. The result is a 610 MB file.

3.2 Improvement of Intermediate Data

For fields of types 1 to 3, we wrote scripts to improve data.

For example:

- Root: Verb root can contain Latin characters among the Georgian characters. They are used to signal either alternatives (for example, "A" means either "ა" or nothing

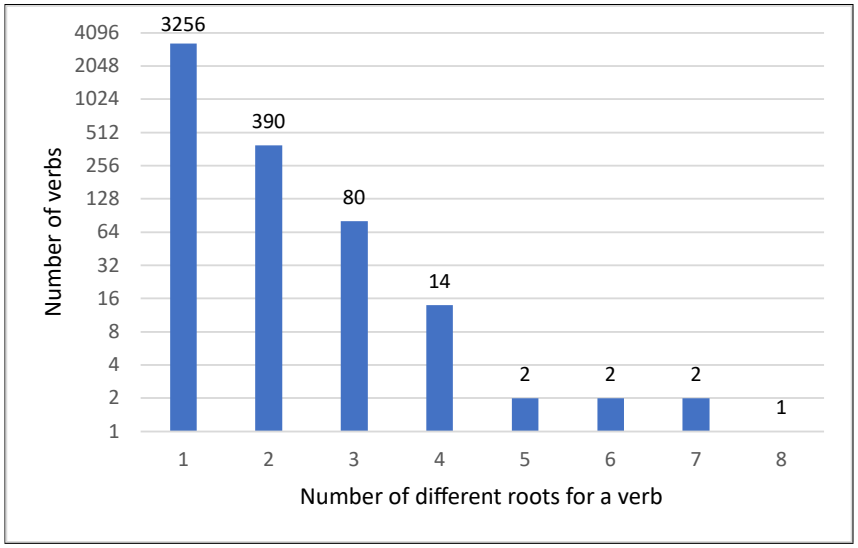

Figure 2: Almost 500 verbs show different roots in their forms

as in ”ვAდ” = ”ვად” (vad) OR ”ვდ” (vd); or strict absence (for example, ”a” means that the ”ა” that may be present in other forms must be absent, as in ”თარგმან” = ”თარგმნ” (targmn)). The script duplicates alternatives by adding new lines and changes Latin characters into either a Georgian character or no character. In addition, Figure 2 shows that verbs may have different roots in their forms. If the vast majority (3256) of the verbs have only one root throughout all their forms, 390 have 2, 80 have 3, 14 have 4, 2 have 5, 6 or 7, and 1 even has 8 roots in the forms of the verbs built. For example: common root ”მბობ” (”mbob”) leads to verbs around the meaning of ”to say” with 7 different roots: ”ამბ” (”amb”), ”თქ” (”tk”), ”თქვ” (”tkv”), ”თხრ” (”tkhr”), ”მბობ” (”mbob”), ”ტყ” (”t’q”), ”უბნ” (”ubn”). Common root ”სვლ” (”svl”) leads to verbs around the meaning of ”to go” and ”to come” with 8 different roots: ”არ” (”ar”), ”დი” (”di”), ”ვედ” (”ved”), ”ველ” (”ved”), ”ვიდ” (”vid”), ”ვლ” (”vl”), ”ს” (”s”), ”სვლ” (”svl”). For a given verb, Clarino gives all the possible roots attached to a common root. However, a given form only contains one of them. The script eliminates all the irrelevant roots from form descriptions.

- Preverb: some forms start with a preverb that gives an indication similar to English postpositions. There are more than 10 possible preverbs. For example, ”ა” (a) conveys the same idea as ”up”. This information is especially crucial to understand Georgian conjugation. It is split into 3 fields in Clarino. If any of the 3 fields is present in a form, the script collects it. If several of the 3 fields are present with different values, the script keeps the value present in the form.
- Verbal noun and Masdar: In section 2 we explained why we must transform Clarino’s verbal noun to generate a masdar. When the verbal noun is available, the script merges it with the preverb. For example, for the form ”გადა-ვა-კეთ-ეთ” (gada-va-ket-et), the preverb is ”გადა” (gada), the verbal noun is ”*კეთება” (*keteba), the deduced ”masdar” is ”გადაკეთება” (gadaketeba);

CSV file

form	tense	person	number	...
ვაადამიანებ	present	1	sg	...
...	...	...	...	...

RDF triplets

<ვაადამიანებ> <tense> <present> .
<ვაადამიანებ> <person> <1> .
<ვაადამიანებ> <number> <sg> .
...

Table 3: Extract from the CSV file and the corresponding RDF triplets

3.3 From CSV To SparkLis

Another Python script converts the CSV format into RDF (Resource Description Framework) Turtle N-triplets to create linked data compatible with Sparklis, the logical information system developed by Ferré (2017) and used in KartuVerbs for navigating in the data. For example, Table 3 shows an extract of the CSV file line and the corresponding Turtle N-triplets entries. Basically, one line of the CSV file with n columns is transformed into $n-1$ triplets of the form "*l_id*" "*p_property name*" "*p_property l_value*". Where "*l_id*" is the content of the first column of line "*l*", "*p_property name*" is the first line of column "*p*" and "*p_property l_value*" is the content of the slot line "*l*" column "*p*".

In order to support the data into two languages, this script also adds transliteration of Latin characters into Georgian characters. The result is a 3.2 GB turtle file. Considering the huge amount of data, it is crucial that the file is indexed for SPARQL to give answers with acceptable response times. Indexation is done with an open-source packages: *apache-jena* and *apache-jena-fuseki*. The result is a 11 GB file. The final step is to create an endpoint for Sparklis, namely to start a SPARQL database server and load the RDF Turtle N-triplets, a standard procedure for Sparklis applications.

4. Decision Tree to Improve Occasional Missing Fields

This section describes the machine learning experiment we made for the improvement of the verbal noun field that is of type 6. Namely, this field should always be present, but it is occasionally absent. Over 600 000 forms (corresponding to 4640 verbs) do not have a verbal noun. Thus, filling up the blanks can be of significant importance. Furthermore, the needed value might already be present in the other records. Indeed, Figure 3 shows that a common-root can lead to multiple verbs. We can see that only 670 common roots lead to a single verb. On the other end of the range, 1 common root leads to 153 different verbs. A common-root leads in average to 9 different verbs. Even if not all verbs coming from a common root have the same verbal noun, the root is crucial to build a verbal noun. In addition, there is only one common root without any information about verbal noun.

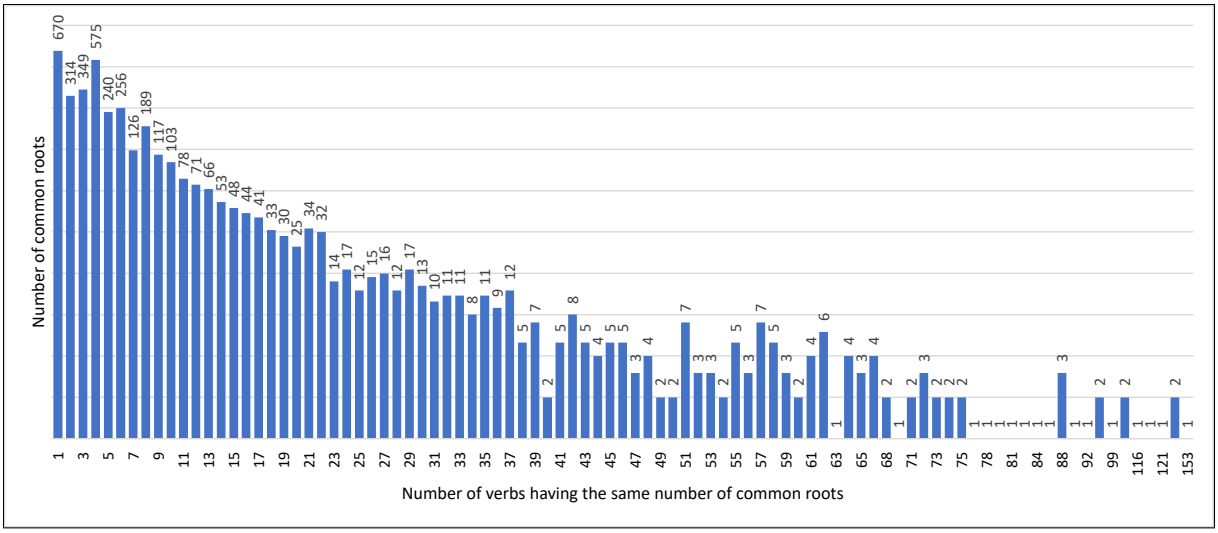

Figure 3: A common root can lead to several verbs - up to 153, 9 in average

The research hypothesis is that there is enough information in the input dataset to predict the missing verbal nouns by machine learning.

The remaining of this section, introduces the experimental setting (input data and training process), presents the results, discusses them and presents some implementation issues.

4.1 Experimental setting

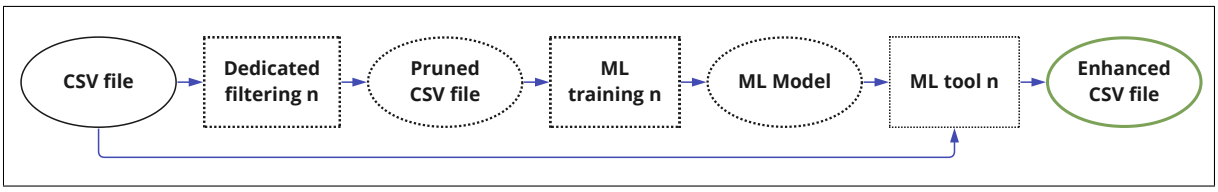

Figure 4: Structure of Machine Learning Tool n

Our improvement process is incremental and, when applying a given tool, not all forms are necessarily reliable. Thus, as illustrated in Figure 4, when using machine learning we first filter out all forms for which a doubt still exists. In particular, for this experiment, all lines without verbal noun have been filtered out before training. After filtering, the input file consists of 3.8 million lines, each one containing a Georgian inflected verb form and 14 of its features. We then train a model, and from this model we generate an enhanced CSV file that can be enhanced further by other tools.

Input Data As discussed in Section 2, fields of type 5 may exhibit an absence of value and it is not necessarily an error. Therefore, forms with such missing values have to be kept in the training data. Missing values, however, have an impact on the chosen machine learning technique (see discussion below). Figure 5 shows the percentage of missing values for these fields. Fields Ending and Preradical have been built from Clarino information by

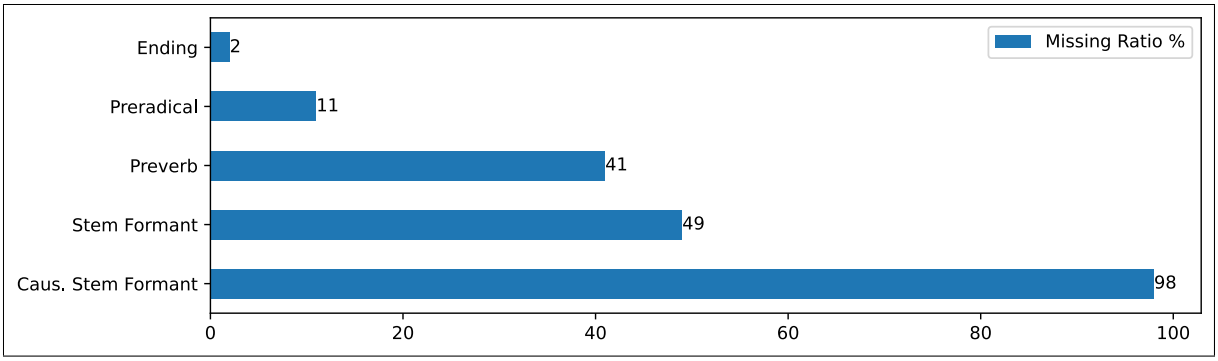

Figure 5: Missing values in input data

scripts not described in this article. Note that Causative Stem Formant is mostly absent. An absence of preverb or stem formant can be perfectly valid for some tenses and verb groups. They are, however, key in the structure of the verbal noun when they exist.

Machine learning algorithms work on numbers. As our fields are mostly symbolic, we had to encode them into numbers. Missing values are encoded by ”0”. Fields with a finite number of possible values (tenses for example) are simply encoded by constants. String fields require more subtle treatments. For some fields, it is sufficient that the encoding is a function (to one string corresponds a single encoding, the same encoding may correspond to several strings). For other fields, it is crucial to build a bijection between string representations and numerical representations in order to be able to interpret the result properly (in a unique way). Here it is crucial to be able to know what the string is suggested by the ML algorithm for missing verbal nouns. It should be noted that missing verbal nouns most probably already exist in other forms in the database. For the fields where a function is sufficient, to encode Georgian characters we refer to the UTF-8 encoding scheme, where Georgian characters are represented by three-byte sequences. The first two bytes are redundant for the conversion process. Thus, to encode a Georgian word into a numeric representation, we extract the last byte from each character and sum their decimal values.

$$f(string) = \sum_{i=1}^{string\_length} Byte\_value(Last\_Byte(UTF-8(character_i)))$$

where $character_i$ is set of individual characters of a Georgian text $string$.

For example, ”გა” ⇒ ”გ”+”ა” ⇒ b’\xe1\x83\x92’ + b’\xe1\x83\x90’ ⇒ b’~~\xe1\x83~~\x92’ + b’~~\xe1\x83~~\x90’ ⇒ value(’b\x92’) + value(’b\x90’) ⇒ 146 + 144 = 290.

However, for verbal nouns, a one-to-one correspondence (bijection) between text and numeric versions of Georgian Verbal nouns is required. Indeed, Verbal Noun is the target variable for our task prediction. With the previous encoding, 290 can be decoded as “გა” and “აგ” as well. Therefore, to each verbal noun we assign a different integer in range [0, 6538] and we keep a correspondence table for the 6539 different verbal nouns.

Training process To train our data we use a supervised learning model, Decision Tree, for the following reasons. It is suited to handle multiclass classification tasks (as discussed in

Bansal et al. (2022)). Our task is, indeed, a classification because a predicted best match of verbal noun should be selected from a set of verbal nouns included in the input file. Furthermore, Decision tree model is non-parametric; before training our model we did not have to determine any parameters. Decision tree algorithm possesses very low complexity. This is crucial considering the size of our input data. Last but not least, Decision Tree model is not influenced by missing values. This is also crucial because our original data contain missing values as illustrated in Figure 5. For the experiment, the filtered database is split into 2 parts: 80% for the training subset and 20% for the testing subset, a typical ratio in data science. We split the dataset using either systematic randomization or a different seed number. Both approaches of splitting led to the same evaluation scores across different runs. More than 10 seed numbers were tried and the resulting scores were the same for all the attempts. The actual verbal nouns were removed from the test dataset and kept for later verification.

4.2 Results and Discussion

	Precision	Recall	F1-score	Support
Accuracy			1.00	759663
Macro avg	1.00	1.00	1.00	759663
Weighted avg	1.00	1.00	1.00	759663

Table 4: Classification report for Decision trees with 14 form characteristics

Table 4 shows the classification report, which assesses the prediction performance for a classification model. The report generates three common metrics that we use to access the quality of the model. Precision is the percentage of correct positive predictions relative to total positive predictions. Calculation: Number of True Positives (TP) divided by the Total Number of True Positives (TP) and False Positives (FP).

$$Precision = \frac{TP}{TP + FP}$$

Recall is the percentage of correct positive predictions relative to total actual positives. Calculation: Number of True Positives (TP) divided by the Total Number of True Positives (TP) and False Negatives (FN).

$$Recall = \frac{TP}{TP + FN}$$

F1 score is a weighted harmonic mean of precision and recall.

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall}$$

The macro-averaged scores are computed by taking the arithmetic means (unweighted means) of all the per-class scores (in our case of all the VN precision scores, recall scores

and f1 scores). This method treats all classes equally regardless of their support values. The weighted-averaged scores (precision, recall and f1 scores) are calculated by taking the mean of all per-class scores while considering each class's support. The 'weight' essentially refers to the proportion of each class's support relative to the sum of all support values. On the Table 4, accuracy refers to micro averaging. It computes a global average F1 score by counting the sums of the True Positives (TP), False Negatives (FN), and False Positives (FP).

$$Accuracy\ F1 = \frac{TP}{TP + \frac{1}{2}(FP + FN)}$$

In multi-class classification cases where each observation has a single label, the micro-F1, micro-precision, micro-recall, and accuracy share the same value (i.e., 1.00 in our case). For each metric, the closer to 1, the better the model. 1 corresponds to 100% of prediction rate.

	Precision	Recall	F1-score	Support
⋮	⋮	⋮	⋮	⋮
813	1.00	0.99	0.99	96
6507	1.00	0.99	0.99	82
4398	0.98	1.00	0.99	62
4094	0.99	1.00	0.99	77
1021	0.97	1.00	0.99	38
6488	0.92	1.00	0.96	12
4882	1.00	0.89	0.94	9
361	0.80	0.87	0.83	113
6453	0.59	0.47	0.52	47
6448	0.00	0.00	0.00	1

Table 5: Classification report for individual verbal noun prediction scores

In Table 4 all scores (macro, micro, and weighted scores) reflect a 100% prediction rate. However, it is important to note that these scores represent averages. Table 5 gives the individual verbal noun prediction scores that are less than 100%, ranging as low as 83%, 52%, and even 0%. The 0% score is due to the fact that there is only one occurrence of verbal noun *ფშვენა (*pshvena, related to a family of verbs around heavily breathing) in the entire training and test datasets. Therefore, there are no other instances for comparison. The 52% corresponds to ბორძიკ (bordzik', to stumble); a verb for which in the training dataset, half of the occurrences have verbal noun ბორძიკ (bordzik') and the other half have verbal noun *ბორძიკება (*bordzik'eba). We applied the trained model to the 600 000 forms with missing verbal noun. We are developing tools to facilitate the validation of the results. We are planning to use a crowd-sourcing platform (see Section 5). We are designing heuristics to reduce the number of results to be manually checked and rank the

results such that experts would be asked to double check the most dubious results first. The heuristics that we have identified so far are,

- A predicted verbal noun is questionable when it does not match the root of the form.
- If forms of the same verb (identified by their Clarino Id) have different verbal nouns these verbal nouns are questionable. It might be the case that all are valid but in that case they should all be attached to all the forms.
- Verbal nouns without a vowel at the end are questionable. Experts can manage with them but not beginners. For example, an expert will understand that verbal noun "ყივლ" (qivl) should be understood as "ყივლი" (qivili, to crow), but a beginner would be lost.
- It is not necessary to check all the forms of a verb. Samples are sufficient, sampling should take into account at least the tense (preferably one that uses a preverb, future for example) and roots. Some verbs exhibit different roots at different tenses or persons.

We have tried the first heuristic combined with the last one, out of the initial the 600 000 forms with missing verbal noun, 100 000 forms have a predicted verbal noun that does not match its root. Taking a sample of these forms resulted in a set of 153 forms that have been checked by hand. Approximately half of them were correct. Although our trained model has achieved a 100% prediction rate, our heuristic observations indicate that the results are not consistently correct. We conjecture that this discrepancy arises from the fact that a limited number of examples in the training data correspond to verbs with a missing verbal noun. Another possibility is that the encoding of Georgian texts utilizes a non-bijective method. Except for verbal nouns, there is no one-to-one correspondence between the Georgian texts and their encoded versions. This unique correspondence presents challenges, as it can result in excessively large and sparse numbers, rendering the machine learning algorithm ineffective or sometimes even impossible to implement.

4.3 Implementation issues

	Virtual Server	Laptop: ROG Zephyrus M16
Model	Intel(R) Xeon(R) Gold 5118 CPU @ 2.30GHz	12th Gen Intel(R) Core(TM) i9-12900H
CPU MHz	2294.612, 32-64 cores	2900.000, 20 cores
Cache size	16896 KB	24576 KB
Memory	64-96 GB	48 GB
Swap Memory	4 GB	400 GB

Table 6: Hardware characteristics

Considering the amount of data of the base (several millions of records), implementation issues are important. Table 6 shows the Hardware characteristics of the experiment. All

the experiments were done in the Linux distributions - Debian 11 (bullseye) and Ubuntu 22.04.2 LTS (jammy). We used free, open-source platform - Python programming language through Jupyter notebook (Anaconda Navigator) and other Unix-tools (awk, sed ...). The Decision tree algorithm is not suitable for variables continuous in nature Bansal et al. (2022). Indeed, using integers instead of floats for verb ID, the F1-score for the predictions went from 77% to 100%. In order to evaluate the performance, we conducted additional tests by training the model on datasets consisting of 10 fields (form, preverb, predarical, root, stem formant, causative stem formant, ending, verb paradigm sub-ID, clarino ID, verbal noun) instead of 15, and 5 fields (form, root, verb paradigm sub-ID, clarino ID, verbal noun). Both datasets yielded similar average results. However, when examining the individual verbal noun prediction rates, the model trained with the larger dataset model outperformed the others. In terms of machine resource consumption and time efficiency, our experiments revealed that there is not a significant disparity between processing 15 fields, 10 fields, and 5 fields. Regardless of the number of fields processed, the model utilized a substantial amount of memory during the prediction phase. Specifically, for our input file, the model required approximately 50 GB of memory, which exceeds the typical memory capacity of machines. To overcome this challenge, we resolved the issue by expanding the SSD-based swap memory. With this configuration in place, our model successfully completed training, testing, and prediction tasks within approximately 2 minutes and 30 seconds for 15 fields, 2 minutes for 10 fields, and 1 minute and 30 seconds for 5 fields input files. Hence, in this particular context, a regular machine or laptop equipped with ample SSD storage can be employed to train extensive datasets using a decision tree algorithm. Although this may lead to a longer processing time, it remains a viable option. We tried another robust model for classification, Support Vector Machine learning model. With only 100,000 rows of input data, and even using maximum cores for parallel computations, it took over 100 times longer than with Decision Tree for the entire data (4 million of lines). It seems impossible to obtain results in a reasonable time for our case.

5. Perspectives

The perspectives are to refine the process and add more improvement tools. We will apply decision tree to occasional incorrect value fields. In Stefanovitch et al. (2022), the authors use machine learning and transformer based models to classify sentiments in Georgian texts. In so doing, they automatically derive all possible morphemes of a verb, based on its root and two additional parameters: a list of potential preverbs, and a dependent noun. We will investigate if their process could be adapted to help improve or validate our morphemes. Another perspective is to investigate how BERT (see Devlin et al. (2019)) could help to add further improvement tools. Language-specific BERT models are not currently available for Georgian. However, there exist multilingual models that include Georgian language, see Wang et al. (2020); Conneau et al. (2019); Pires et al. (2019). Besides conjugation we also plan to use it for different tasks such as morphological tagging and Named Entity Recognition classification, along the lines of the work for Estonian of Kittask et al. (2020). It will enable us to enrich the base with new properties. Of possible interest are also the network structures to learn word embedding, sentence embedding, and sequence generation with transformers like BERT, introduced in Zhou et al. (2020).

We plan to use crowdsourcing in order to give a chance to users and experts to signal mistakes or missing information. The IRISA platform Headwork[2] will be used. Indeed, the Georgian language contains so many exceptions to the conjugation rules that we do not expect machine learning tools, however efficient, to produce 100% correct information.

6. Conclusion

In this paper, we described a process to transform textual structured knowledge into semantic linked data, applied to Georgian verbal knowledge. The target users and the objectives of the two knowledge bases differ. Hence, initial data have to be reconstructed and interpreted to fit KartuVerbs objectives. The described process aims at applying successively a number of improvement tools. A specific one, using decision tree for machine learning, has been described in detail to complement occasional missing values. The average F1-score for the generated model is 100%. The scripts produced so far are freely available on the net [3]. They can be adapted to other applications to help transform data produced for given objectives into other data suited for different objectives.

7. Acknowledgements

We are indebted to Paul Meurer who granted us a private access to a web version of the base behind the Georgian functionalities of https://clarino.uib.no/iness. We thank Mikheil Sulikashvili for his help to scrap Clarino web pages.

This research PHDF-22-1840 is supported by Shota Rustaveli National Science Foundation of Georgia (SRNSFG) and by ANR Project SmartFCA, ANR-21-CE23-0023.

8. References

Bansal, M., Goyal, A. & Choudhary, A. (2022). A comparative analysis of K-Nearest Neighbor, Genetic, Support Vector Machine, Decision Tree, and Long Short Term Memory algorithms in machine learning. *Decision Analytics Journal*, 3, p. 100071. URL https://www.sciencedirect.com/science/article/pii/S2772662222000261.

Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzman, F., Grave, E., Ott, M., Zettlemoyer, L. & Stoyanov, V. (2019). Unsupervised Cross-lingual Representation Learning at Scale. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. Online: Association for Computational Linguistics*, pp. 8440–8451. URL https://aclanthology.org/2020.acl-main.747.

Devlin, J., Chang, M.W., Lee, K. & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). Minneapolis, Minnesota: Association for Computational Linguistics*, pp. 4171–4186. URL https://aclanthology.org/N19-1423.

Ducassé, M. (2020). Kartu-Verbs: A Semantic Web Base of Inflected Georgian Verb Forms to Bypass Georgian Verb Lemmatization Issues. *Z. Gavriilidou, M. Mitsiaki &*

[2] https://druid-garden.irisa.fr/spipollhw/

[3] https://github.com/aelizbarashvili/KartuVerbs

A. Fliatouras (eds.) Proceedings of XIX EURALEX International Congress, volume 1. SynMorPhoSe Lab, Democritus University of Thrace, pp. 81–89. URL https://euralex2020.gr/wp-content/uploads/2020/11/EURALEX2020_ProceedingsBook-p081-089.pdf.

Ducassé, M. & Elizbarashvili, A. (2022). Finding Lemmas in Agglutinative and Inflectional Language Dictionaries with Logical Information Systems: The Case of Georgian verbs. *Proceedings of XX EURALEX International Congress*, p. 6.

Ferré, S. (2017). Sparklis: An Expressive Query Builder for SPARQL Endpoints with Guidance in Natural Language. *Semantic Web: Interoperability, Usability, Applicability*, pp. 405–418. URL http://www.semantic-web-journal.net/content/sparklis-expressive-query-builder-sparql-endpoints-guidance-natural-language-1.

Kittask, C., Milintsevich, K. & Sirts, K. (2020). Evaluating multilingual BERT for Estonian. *A. Utka et al. (ed.) Human Language Technologies – The Baltic Perspective. IOS Press Online.*

Meurer, P. (2007). A computational grammar for Georgian. *International Tbilisi Symposium on Logic, Language, and Computation. Springer*, pp. 1–15.

Pires, T., Schlinger, E. & Garrette, D. (2019). How multilingual is Multilingual BERT? *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 4996—5001.

Stefanovitch, N., Piskorski, J. & Kharazi, S. (2022). Resources and Experiments on Sentiment Classification for Georgian. *International Conference on Language Resources and Evaluation*, pp. 1613–1621.

Tschenkéli, K. (1965). Georgisch-deutsches Wörterbuch, volume 2. *Amirani-Verlag Zürich.*

Tuite, K. (1998). Kartvelian morphosyntax: Number agreement and morphosyntactic orientation in the South Caucasian languages. *Lincom Europa Munich.*

Wang, Z., Mayhew, S., Roth, D. et al. (2020). Extending Multilingual BERT to Low-Resource Languages. *Findings of the Association for Computational Linguistics: EMNLP 2020. Online: Association for Computational Linguistics*, pp. 2649–2656. URL https://aclanthology.org/2020.findings-emnlp.240.

Zhou, M., Duan, N., Liu, S. & Shum, H.Y. (2020). Progress in Neural NLP: Modeling, Learning, and Reasoning. *Engineering*, pp. 275–290. URL https://www.sciencedirect.com/science/article/pii/S2095809919304928.