Learning How to Conjugate the Romanian Verb Rules for Regular and Partially Irregular Verbs

I. Stem alternations in Romanian verbs

Stem alternations, or apophony, is one of the reasons why the Romanian language is difficult to acquire.

For partially irregular verbs it is not enough to learn a generic suffix variation pattern, because there are simultaneous variations in the stem.

Conjugation of "a purta":
eu port(ă) (you wear)
eu poartă (you wear)

Conjugation of "a curta":
eu curtes(ă) (you count)
eu curtesă (you count)

What is needed for automatic conjugation?

Romanian received a Latin-inspired classification of verbs into 4 conjugational classes, based on the ending of the infinitive form. This does not discriminate the two verbs shown above, so the standard model is insufficient.

The goal: given an infinitive form, know what letters change, and how they change.

The trick: craft a sufficient, near-exhaustive, disjoint set of conjugation classes.

II. Previous work

Moisil (1960): variable letters

\[
\text{purta} \Rightarrow \text{purita}, \text{purita} \text{ where: } \text{u} = \{u, o, a\}, \text{t} = \{t, f\}
\]


Dinu et al (2011): 7 conjugation classes for verbs ending in -ta. Knowing the class means knowing the alternations that occur. Idea: the classes can be learned.

A class corresponds to a conjugation rule: a set of 6 regular expressions matching the 6 conjugation forms of present tense verbs. Parts of the forms that are not accounted for must remain fixed, i.e. a rule accounts for all the variation.

Classification using character n-gram features + SVM: n-gram size chosen to be 3 for model simplicity (n = 5 is optimal).

Input: "purta" \(\Rightarrow\) \(\{u, o, a, \text{p}, \text{u}, \text{ur}, \text{e}, \text{t}, \text{a}, \text{pum}, \text{ure}, \text{t}\} \)

Output: label in \(\{1, 2, 3, 4, 5, 6, 7\}\), imbalanced classes

Results: 82.71% accuracy and 80% F-score

III. Crafting conjugation rules using regular expressions

Process
To manually expand a set of conjugation rules:
1. Select unmatched verb
2. Add rule to completely conjugate it
3. Match verbs against new rules

For example:
1. Say the verb 'a omori' (to kill) is not matched.
2. A conjugation rule matching this verb would be:
   1. \(s\): \(\ast\) \(\ast\) \(\ast\) \(\ast\)\(\ast\) \(\ast\) \(\ast\) \(\ast\) omor
   2. \(s\): \(\ast\) \(\ast\) \(\ast\) \(\ast\)\(\ast\) \(\ast\) \(\ast\) \(\ast\) omor
   3. \(s\): \(\ast\) \(\ast\) \(\ast\) \(\ast\)\(\ast\) \(\ast\) \(\ast\) \(\ast\) omor
   4. \(s\): \(\ast\) \(\ast\) \(\ast\) \(\ast\)\(\ast\) \(\ast\) \(\ast\) \(\ast\) omor

3. This rule also matches, among others, the verb 'a dobori' (to defeat) so mark this one as matched too.

IV. Classification methodology

Dataset of Romanian verb forms

Extract indicative present tense, label infinitives

Add suffix marker 'purta' -> 'purita'$

Extract n-grams up to size $n$

Vectorize into n-gram frequency or occurrence (binarized) vectors

Classify using Linear SVC

Estimate scores using 10-fold cross validation

V. Conclusion and perspectives

Estimated scores:

Parameters chosen by grid search: n=5, append '$', do not binarize, C=0.1

Correct classification rate: 90.64% (baseline choosing most probable class: 48%)

Weighted averaged precision: 80.90%, recall: 90.64%, F1 score: 89.89%

Appending the artificial terminator marker '$' consistently improves accuracy by around 0.7% irrelevant of the other parameters.

Frequency features perform slightly better than binarized ones for this task.

What does this mean?

Verb conjugation can be learned with good scores, even with the assumption that classes do not interact. Our classes are coarse-grained. An exhaustive model, at least for the training data, will need to have many classes for unique and near-unique conjugation patterns. For better generalization we need a finer-grained system.

Future work and collaboration ideas:

Build a more compact model by eliminating rule interactions (see discussion above)
Compare with hand-crafted rule based conjugation
Try human evaluation on unseen, untagged verbs
Actually build a verb conjugation using classification output (trivial)
Extend to other languages with similar behaviour (Hebrew)