

MULLE: A grammar-based Latin language learning tool to supplement the classroom setting

Herbert Lange and Peter Ljunglöf

Computer Science and Engineering

University of Gothenburg and Chalmers University of Technology

herbert.lange@cse.gu.se peter.ljunglof@cse.gu.se

Abstract

MULLE is a tool for language learning that focuses on teaching Latin as a foreign language. It is aimed for easy integration into the traditional classroom setting and syllabus, which makes it distinct from other language learning tools that provide standalone learning experience. It uses grammar-based lessons and embraces methods of gamification to improve the learner motivation. The main type of exercise provided by our application is to practice translation, but it is also possible to shift the focus to vocabulary or morphology training.

1 Introduction

Computer-assisted language learning is a growing field that is also more and more in the focus of the general public thanks to popular tools such as Duolingo¹ or Rosetta Stone.² In combination with the rise of the smartphone it has become possible to learn languages almost any time and anywhere in an entertaining way.

Text input on mobile devices equipped with touch screens as the primary input device can be difficult, but is relevant to language learning tasks. This general problem led to the development of several alternative input methods (Ward et al., 2002; Kumar et al., 2012; Felzer et al., 2014; Shibata et al., 2016) including Ljunglöf's method of grammar-backed word-based text editing (2011).

We present the MUSTE³ Language Learning Environment (MULLE)⁴, an application for lan-

guage learning that combines several techniques: tree-based sentence modification, controlled natural language grammars for the exercise creation as well as concepts of gamification.

The goal of our system is to provide a tool that enriches the traditional language learning setting in an enjoyable way and helps to avoid problems with learner motivation that can be encountered in language classes.

2 Previous and related work

MULLE is based on an underlying theory of word-based grammatical text editing by Ljunglöf (2011).

The software used to translate between the surface text and the syntax trees is the Grammatical Framework (GF) (Ranta, 2009b, 2011). It is a grammar formalism and parsing framework based on type theory. On top of this formalism, a multilingual library of grammars is build, the so-called Resource Grammar Library (RGL) (Ranta, 2009a) which covers more than 30 languages including Latin (Lange, 2017). It provides an interface that can be used to implement more application-specific grammars similar to an Application Programming Interface (API) in computer programming.

An important aspect of CALL is the factor of both long and short-term motivation for which the concept of gamification is relevant (Deterding et al., 2011). Several approaches are possible, of which we focus on GameFlow by Sweeter and Wyeth (2005) and MICE by Lafourcade (described in Fort et al. (2014, section 4)). GameFlow translates the more general Flow approach (Csikszentmihalyi, 1990) to computer games.

Finally, comparison to other language systems is relevant for our work. Most language systems share common features, especially translation ex-

¹<https://www.duolingo.com/>

²<http://www.rosettastone.eu/>

³<http://www.cse.chalmers.se/~peb/muste.html>

⁴<https://github.com/MUSTE-Project/MULLE>

ercises seem quite similar across different systems. Still there are major differences in the way these systems work and the use cases they are developed for. Duolingo for example heavily relies on text input created by the user, uses a mixture of user-generated content and machine learning techniques (Horie, 2017) and is meant for open independent online learning mostly for modern languages. MULLE on the other hand uses resources created by experts, does not require text input created by the user, and is intended for, but not restricted to, accompanying language classes in a closed classroom setting.

3 Creation of interactive exercises from a Latin textbook

The idea of grammar-based text modification led us to the creation of MULLE. It is game-like and the player solves language learning exercises focusing on translation. Each exercise consists of two sentences in different languages, one language that the user already knows (i.e. the meta-language), and the language to be learned (i.e. the object language). Both sentences differ in some respect, depending on the grammatical features that the lesson is focusing on.

Using GF together with the RGL helps us to create domain-specific grammars in a straightforward way. Such grammars can be designed to catch exactly the complexity of the lessons in a classic textbook. That way we can mirror the same lesson structure in MULLE, at the same time adding more flexibility and giving the possibility of generating a large supply of interactive exercises with plenty of variation using vocabulary and concepts familiar from class and textbook.

A textbook for language learning is usually split into a sequence of lessons with texts and exercises of growing syntactical complexity. This is the case for textbooks both at high-school and university levels (e.g. Lindauer et al. (2000); Ehrling (2015)). Typically, each chapter consists of a text part, a vocabulary list, some grammatical explanation and additional exercises. The growing vocabulary and increase in complexity helps the student learn the whole of a language in a slow pace. This approach is also common in language learning applications and can readily be implemented in MULLE.

Each grammar lesson in MULLE covers a set of interactive exercises. So we need lesson-specific

grammars that use the same lexicon and grammatical constructions as the corresponding parts of the textbook. For that we can use the RGL, when writing a new grammar for a lesson we already have access to an extensive description of the languages we want to cover and only have to select the concepts we want to include.

First a lexicon is created that covers exactly the vocabulary of a lesson. Extensive lexical resources are already available for GF and they can easily be extended by the author of the grammar relying on the morphological component of the grammar to generate the correct word forms.

Next the grammatical constructions that will be used in this lesson are selected by exposing only the parts that are relevant to the planned learning outcomes. The RGL can be seen as a collection of grammatical constructions, and each lesson uses a subset of these concepts. So by only providing a restricted subset together with the selected vocabulary it is possible control the complexity of the lessons.

Finally every grammar we create needs to be multilingual for at least two languages: the meta-language (e.g. Swedish), and the object language (e.g. Latin). Since the RGL is inherently multilingual it is straightforward to provide the lessons in multiple languages; With only minimal adjustments we can cover as many languages as we want as long as they are already included in the RGL.

The usual size of the lesson grammars we encountered so far was between 50 and 100 lexical items and about 20 syntax rules.

The main focus of our work is on one form of translation exercises but other forms of exercises are also useful in the context of language learning. That usually includes explicit vocabulary exercises and, in the case of languages with a strong morphology like Latin, some exercise for practicing word forms.

Practicing vocabulary is possible either by using lexical categories as top-level categories of the syntax trees or by using sentences that are almost correct except for a lexical mismatch in one position.

Exercises for morphology involve slightly more work since our grammar formalism by default only creates grammatical sentences including correct word agreement. So to be able to practice morphology in our setup have to relax these morphological constraints in the grammars. That gives

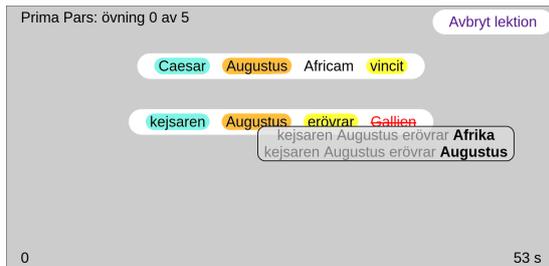


Figure 1: Screenshot of the exercise view

us a way to create exercises where the user has to both identify wrong morphological forms in a sentence and find the right form to replace them with.

4 Implementation

Based on these ideas we have implemented MULLE which can already be used in language classes. In order to be independent of certain kinds of devices and operating systems we provide the whole application as a browser-based online application.

The application is developed independent of the grammars that can be used. That means that the whole system can be set up by providing the application with a set of lesson grammars and a fully usable language learning environment is available.

4.1 User interface

The user interface is kept minimalist, as can be seen in Figure 1, and only provides the user with the most essential information, including the current score count, the sample sentence in the metalanguage and the modifiable sentence in the object language that has to be altered to match the sample, and the time elapsed since starting the exercise as well as clicks spent on the exercise.

Colours are an important aspect of the interface because they indicate progress in solving the exercise. The background colours of the words highlight which parts of the two sentences already match up with each other. In the example “kejsaren” is a proper translation of “Caesar” which is shown by highlighting them in the same colour. The same is the case for both occurrences of “Augustus” as well as the pair “vincit” and “erövrar”. The meaning of the colours is that phrases in the same colour are translations of each other. Only one pair of words, “Africam” and “Gallien”, is not highlighted, so here some user intervention is needed.

This current design reduces the possible distractions while supporting the learner. Depending

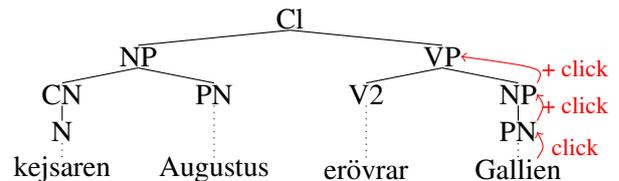


Figure 2: Syntax tree including the path through the tree after several clicks on the word “Gallien”

on the target age group a more elaborate graphics design could have a more positive effect on the acceptance of the system.

4.2 Gamification

We presented two approaches for gamification in Section 2, based on which we selected certain aspects to be included in our application. For our application the following features of GameFlow seem most relevant: *Concentration*, i.e., minimising the distraction from the task, *Challenge* by giving a scoring schema, *Control* by providing an intuitive way to modify the sentence, *Clear goals* by providing a lesson structure, and *Immediate feedback* with the colour schema.

The concept of lessons and exercises is essential for this kind of language learning because it makes the learning progress explicit. The completed lessons are presented to the student together with the scores, so that they can see their own progress on the way to reaching their final goal of learning the language.

By applying methods from GameFlow, we positively influence the motivation while learning a new language. Adding more features of gamification, especially involving social aspects, is a possible extension for the future.

5 User interaction

After logging into the system the user is presented with a list of lessons and the current status, i.e. the number of finished exercises for each lesson and the current score. Some lessons might be disabled because they require previous lessons to be completed first. Now the user can choose one of the enabled lessons to start the exercises.

As soon as a user starts a lesson a set of exercises is selected. These exercises are chosen from a list of exercises in a database. The exercises consist of two syntax trees that differ in certain grammatical aspects. Associated with each syntax tree is one sentence, one in the metalanguage and one in the object language. The syntax trees are

hidden from the user and only implicitly influence the user experience.

The exercises are presented in the form shown in Figure 1. The background colours of the words show the state of the translation. When the user clicks on one of the words in the bottom sentence, they are presented with a list of potential replacements. This selection is based on where in the tree the word is introduced. In the example the user clicked on the word “Gallien”, which is a proper name, so all proper names contained in the grammar are presented. By clicking several times on the same word the focus can be expanded to cover larger phrases, e.g. from proper name to noun phrase, and so on, by traversing upwards through the tree (Figure 2). The menu contains all phrases of the syntactic category selected by clicking on words. That means that suggestions can contain more or less words than currently in focus. So for example if a noun phrase is in focus, both noun phrases with and without adjectives appear in the list. Selecting a longer phrase is the same as inserting words in the sentence and selecting shorter phrases corresponds to deleting words from the phrase.

With these operations, i.e. substitution, insertion, and deletion, the user can modify the sentence to finish their task. When the two sentences are proper translation of each other, i.e. the two syntax trees are similar, the user is congratulated on the success and presented the final score.

Lessons can be interrupted and resumed at any time as well as repeated to improve the score.

6 Evaluation

For the evaluation of our approach we have designed an experiment setup. The full setup includes a basic placement test in the beginning that is repeated at the end of the test period to provide information about the learning outcome. The placement test consists of a fixed set of exercises from all lessons that will be covered during the experiment period. Both error rate and completion time are measured. A questionnaire controls for factors like learner background, previous knowledge, etc. It also gives insight into the learner motivation in the beginning so it can be repeated in the end to see any development in this relevant aspect. Then over the span of the experiment the students can use the software independently online. The lessons are kept in sync with

the syllabus of the course that is accompanied by the experiment. In the end the collected data consists of changes in learning outcome and learner motivation as well as activity of the student in the system.

In a pilot experiment we tried aspects of this experimental evaluation. The results were not yet statistically significant because the course size was very small and the dropout rate was high. From the initial 10 Students only 4 finished the course so we only received complete feedback from two students out of initially 6 participants. Anyways, the general interest, both by teachers and students, in this kind of application is strong.

A larger scale follow-up experiment will focus on the change in the learner attitude, which is relevant for showing that our tool is suited for tackling potential anxiety in learners, a problem Latin teachers have pointed out (Dimitrijevic, 2017). With more participants different kinds of control and test conditions can be introduced.

7 Discussion

One challenge with the user interface is the semantics of clicks, especially concerning word insertion. Clicking on a gap between two words to insert words seems more intuitive than clicking on a word. But where to click might also depend on the languages involved.

Another important question for the current application is the influence of the grammar design both on the learning experience and the learning outcome. It is possible to vary the design of the grammar to change the behaviour of our system.

Related is the role of semantics in the lesson grammars. The lessons and exercises are meant for learning the syntax of a language but non-sensical semantics can be an obstacle for the learning process. For example the famous sentence “Colorless green ideas sleep furiously” (Chomsky, 1957, p. 15) is considered grammatical but would probably distract the learner.

8 Future work

This project is work in progress and we plan to extend the system in several ways. First, we will repeat the experiment from Section 6 on a larger scale. Furthermore we plan to extend our implementation to become more feature-rich with a special focus on investigating the points addressed in the discussion section. Finally we want to con-

tinue collaborating both with teachers and students to improve the system in order to enrich teaching and learning Latin.

References

- Noam Chomsky. 1957. *Syntactic Structures*. Mouton de Gruyter, Berlin New York. Reprint 2002.
- Mihaly Csikszentmihalyi. 1990. *Flow: The Psychology of Optimal Experience*. HarperCollins.
- Sebastian Deterding, Dan Dixon, Rilla Khaled, and Lennart Nacke. 2011. From Game Design Elements to Gamefulness: Defining "Gamification". In *Proceedings of the 15th International Academic MindTrek Conference: Envisioning Future Media Environments*, pages 9–15, New York, NY, USA. ACM.
- Dragana Dimitrijevic. 2017. Latin Curricula, Attitudes and Achievement: An Empirical Investigation. In *Proceedings of the 19th International Colloquium on Latin Linguistics*.
- Sara Ehrling. 2015. *Lingua Latina novo modo – En nybörjarbok i latin för universitetsbruk*. University of Gothenburg.
- Torsten Felzer, Ian Scott MacKenzie, and Stephan Rinderknecht. 2014. Efficient computer operation for users with a neuromuscular disease with On-ScreenDualScribe. *Journal of Interaction Science*, 2(2).
- Karën Fort, Bruno Guillaume, and Hadrien Chastant. 2014. Creating Zombilingo, a Game with a Purpose for Dependency Syntax Annotation. In *Proceedings of the First International Workshop on Gamification for Information Retrieval, GamifIR '14*, pages 2–6, New York, NY, USA. ACM.
- André Kenji Horie. 2017. Rewriting Duolingo's engine in Scala. <http://making.duolingo.com/rewriting-duolingos-engine-in-scala>. Accessed 04.04.2018.
- Anuj Kumar, Tim Paek, and Bongshin Lee. 2012. Voice typing: A new speech interaction model for dictation on touchscreen devices. In *Proceedings of CHI 2012, SIGCHI Conference on Human Factors in Computing Systems*, Austin, Texas, USA.
- Herbert Lange. 2017. Implementation of a Latin Grammar in Grammatical Framework. In *DATECH2017*, Göttingen, Germany.
- Josef Lindauer, Klaus Westphalen, and Bernd Kreiler. 2000. *Roma, Ausgabe C für Bayern, Bd.1*. C.C. Buchner.
- Peter Ljunglöf. 2011. Editing Syntax Trees on the Surface. In *Nodalida'11: 18th Nordic Conference of Computational Linguistics*, Rīga, Latvia.
- Aarne Ranta. 2009a. The GF Resource Grammar Library. *Linguistic Issues in Language Technology*, 2(2).
- Aarne Ranta. 2009b. Grammatical Framework: A Multilingual Grammar Formalism. *Language and Linguistics Compass*, 3(5):1242–1265.
- Aarne Ranta. 2011. *Grammatical Framework: Programming with Multilingual Grammars*. CSLI Publications, Stanford.
- Tomoki Shibata, Daniel Afergan, Danielle Kong, Beste F. Yuksel, I. Scott MacKenzie, and Robert J.K. Jacob. 2016. Text Entry for Ultra-Small Touchscreens Using a Fixed Cursor and Movable Keyboard. In *Proceedings of CHI 2016: The ACM SIGCHI Conference Extended Abstracts on Human Factors in Computing Systems*, Santa Clara, California.
- Penelope Sweetser and Peta Wyeth. 2005. GameFlow: A Model for Evaluating Player Enjoyment in Games. *Comput. Entertain.*, 3(3):3–3.
- David J. Ward, Alan F. Blackwell Y, and David J. C. Mackay Z. 2002. Dasher: A Gesture-Driven Data Entry Interface for Mobile Computing Human-Computer Interaction. *Human-Computer Interaction*, page 228.