Originally published in: Davis, Brian/Keet, C. Maria/Wyner, Adam (Eds.): Controlled Natural Language. Proceedings of the Sixth International Workshop, CNL 2018, Maynooth, Co. Kildare, Ireland, August 27–28, 2018. – Amsterdam: IOS Press, 2018. Pp. 61-70. (Frontiers in Artificial Intelligence and Applications 304)

doi:10.3233/978-1-61499-904-1-61

Putting Control into Language Learning

Herbert LANGE [a] Peter LJUNGLÖF [a]
[a] *Computer Science and Engineering*
University of Gothenburg

Abstract Controlled Natural Languages (CNLs) have many applications including document authoring, automatic reasoning on texts and reliable machine translation, but their application is not limited to these areas. We explore a new application area of CNLs, the use of CNLs in computer-assisted language learning. In this paper we present a a web application for language learning using CNLs as well as a detailed description of the properties of the family of CNLs it uses.

1. Introduction

Controlled Natural Languages (CNLs) are an active field of research in Computational Linguistics. Their definition usually is vague, but the general consensus is that they are constructed languages placed somewhere between full natural languages on the one hand and formal languages on the other. Kuhn discusses the definition in [1] and presents typical applications for CNLs, such as machine translation and document authoring. He also expects other applications without going into detail about these possibilities. In this paper we present one new application of CNLs, the use in computer-assisted language learning (CALL).

We present the MUSTE Language Learning Environment (MULLE)[1], a tool for learning languages as a second or foreign language which can use CNLs for the description of learning objectives, and automatically generate translation exercises from formal grammars of these languages. The application itself uses methods similar to conceptual authoring [2] to edit sentences in natural languages in order to make them proper translations of each other. By using CNL grammars as the basis for the exercises, and using textbook lessons as the basis of the grammars, it is possible to create a learning environment that complements the traditional classroom setting.

This article is structured as follows: In Section 2 we present related work, both in CNLs and CALL. Section 3 describes the language learning application we designed. It describes the interface provided to the user and gives details about the grammars used. In Section 4 we sketch an experimental method to evaluate our system and describe a small-scale pilot. Finally in Section 5 we discuss further questions and conclude the article in Section 6 with a look to possible future work.

[1] `https://github.com/MUSTE-Project/MULLE`

2. Related Work

The use of CNLs for language learning seems to be a new application in this field that has not yet been broadly discussed. For this reason, the amount of directly related work is rather limited with the exception of [3]. However, this application can also be viewed as a combination of two tasks that have been popular among the CNL community: reliable machine translation and user support for text edition and creation.

2.1. Related CNL work

Quite often the motivation for the use of Controlled Natural Language is their proximity to formal language. This allows, e.g. automatic reasoning within and reliable translation of these languages. To guarantee that the language used by an author is covered by a CNL, special editors have been proposed. Two approaches are conceptual editing [2] and predictive editing [4,5]. Predictive editing uses chart parsers and compatible grammars to suggest only valid continuations in the process of writing. Some systems only support a fixed vocabulary while other systems support the extension of the vocabulary while the document is authored.

Conceptual editing instead refines the underlying representation by manipulating the surface presentation, i.e. the natural language sentence. In this process so called "holes" are created and filled.

Angelov and Ranta [5] not only present a predictive editor, but also suggest a translation system based on Attempto Controlled English (ACE) [6]. It provides reliable machine translation via abstract syntax trees in the Grammatical Framework (GF) [7,8].

Some CNLs were designed to aid learning languages at a time before computers were considered a tool for it. One example is Basic English [9], an auxiliary language created to help people learn English as a second or foreign language, invented at the beginning of the 20th century.

2.2. Related work within language learning

In the field of language learning applications several approaches can be observed. They range from finite-state technology for morphology training [10] over annotated text data, or semantic resources combined with rule-based algorithms [11,12,13] to user-generated content in combination with machine learning [14]. The aim and scope of theses systems also varies broadly, including the learning environment they target. The systems [12,13] target a closed classroom setting with specific language classes while [10,11] aim at a broader learning environment and are applicable outside a specific course. Modern general-purpose systems like Duolingo[2] target independent language learners.

Reliability also varies between these systems. The smallest scale systems provide the most reliable examples while the most general systems being the least reliable.

[2] `https://www.duolingo.com`

3. Application: Language Learning using CNLs

In this paper we describe a web-based language learning tool which takes advantage of CNL grammar features. It is intended for use in a closed, classroom-related learning environment and provides reliable translation exercises by using fully formalised grammars. The tool presents exercises grouped into lessons where the user is presented with sentences in two different languages.

Usually two languages are involved in language learning: one language the user already knows is used for instructions in the language classes (the meta language) and one language the user is learning (the object language) by discussing it in the meta language . So the meta language and object language in the classroom determine the two languages used in our exercises.

The task of the user is to edit one of the sentences to make it a proper translation of the other. Currently, this means that the underlying GF abstract syntax trees for both sentences have to be the same. As a future development, we imagine an extension where we consider not single trees but sets of trees, in order to be able to handle ambiguous parses. In this case it would be sufficient to have at least one abstract tree in the intersection of the two sets.

The tool has been used in an introductory course in Latin for Swedish students. So in all examples given here Swedish acts as the meta language and Latin as the object language. We implemented the first four lessons of [15] and conducted the pilot of a user study that is described in more detail in Section 4.

3.1. The editing interface

Our application uses a method for word-based text-editing [16], which is in principle related to conceptual editing. It uses syntax trees as formal representations and provides editing operations like insertion, deletion, and substitution on the surface by mapping them onto tree operations. In our application we only look at complete syntax trees, that means we do not use "holes" for incremental creation, but instead modify complete syntax trees.

The editing operations work in the following way: the user clicks on a word in the sentence or on the gap between two words in the sentence. The click position is translated to the node in the syntax tree in which the word is introduced or the closest node covering the space that was clicked. Based on the subtree below this node, all subtrees with the same root category are computed and their linearisations, i.e. their surface representations, are collected and presented to the user.

To clarify this process a set of screenshots with the corresponding syntax tree can be seen in Figures 1–5. In the screenshot one can see the two sentences in different languages. The sentence at the top is fixed and the sentence at the bottom can be changed by the user. The syntax tree beside the screenshot describes the sentence at the bottom. Figure 1 shows the start of the system before any click. Clicks on words in the sentence are then translated to pointers into the tree. For example clicking on the word "Gallien" will set the pointer (circled node) to the node introducing this string, in this case the rightmost PN node (Figure 2). Then the category in the focus of the pointer is used to compute all similar trees of

```
Prima scripta Latina

[...] Imperium imperatorem habet. Imperator imperium tenet. Caesar
Augustus imperator Romanus est. Imperium Romanum tenet. Multas
civitates externas vincit. Saepe civitates victae provinciae deveniunt.
[...]
```

Figure 6. Sample from the text fragment in the first lesson in [15]

with the same category in the root. These trees are used to suggest potential replacements which are shown to the user in the form of a menu. Clicking on the same word several times moves the pointer up in the tree which changes the root category of the candidate trees from PN to NP, VP and so on, and suggests different changes to the sentence (Figure 3–4) before finishing in the root of the tree with category Cl (Figure 5) where the menu does not change anymore. In this way larger phrases can be changed at the same time of more global features like sentence negation can be modified. When instead the user clicks on a different position than before, then the system is reset and the pointer again points to the node indicated by the new click.

3.2. Lessons and exercises

The application provides of a set of lessons, each consisting of a number of exercises. A lesson is defined by a multilingual GF grammar which is derived from a part of a textbook. These parts usually consist of text fragment (a sample can be seen in Figure 6), a vocabulary list, some explanation of grammatical phenomena, as well as some exercises that are supposed to be solved on paper.

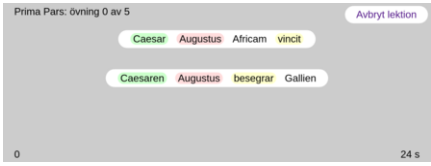
We adopt this structure in our system by using the same text fragments presented in the textbook lessons and formalising them in separate grammars that cover the vocabulary as well as the syntactic constructions used in the corresponding parts of the textbook.

Given a lesson grammar, an exercise consists of two syntax trees and their surface representations. With the task described before a score is calculated based on both the number of clicks and the time spent before finishing the exercise. After finishing a certain number of exercises, the lesson is considered finished.

3.3. Creating the lesson grammars

We create a GF grammar for each textbook lesson in the following way. This work should be automated as much as possible but the basic procedure can also be executed manually.

1. The first step is to adapt a lexicon for the textbook lesson, which is given as an explicit vocabulary list in the book. We can use existing reliable lexical resources or GF smart paradigms [17] to implement it in our grammar.
2. The next step is to create syntax trees for all sentences in the text (Figures 7–8). This can either be done manually or semi-automatically. To automate this process, we parse each sentence using the GF resource gram-



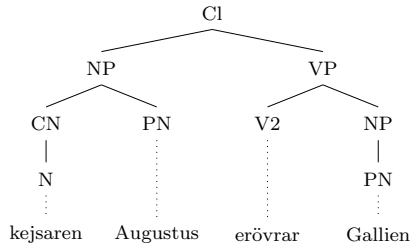
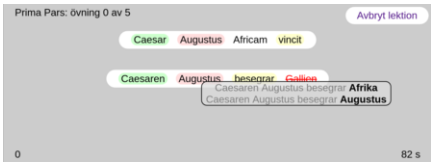


Figure 1. System before any click



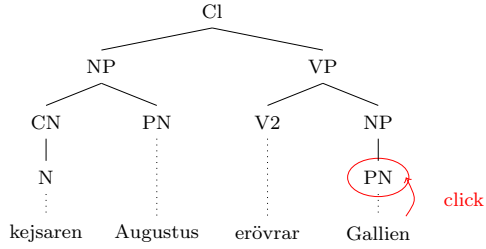
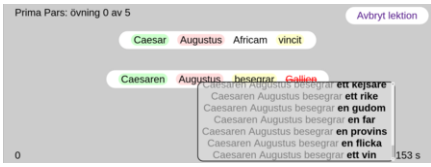


Figure 2. System after one click on "Gallien"



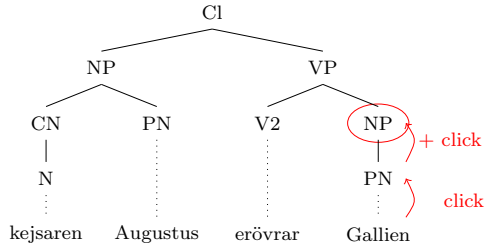
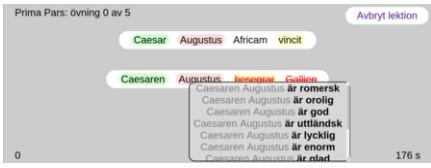


Figure 3. System after second click on "Gallien"



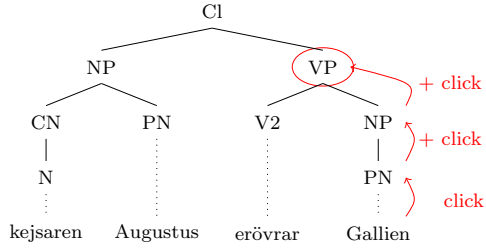
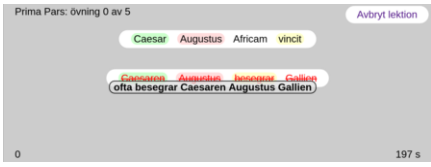


Figure 4. System after third click on "Gallien"



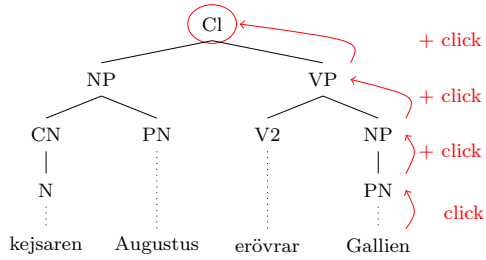


Figure 5. System after fourth click on "Gallien"

mar library [18] augmented with the lexicon from step 1. Because of syntactic ambiguities this might result in several possible trees, so afterwards we have to manually select the correct syntax tree, i.e. the desired analysis of the sentence. This involves some linguistic knowledge from the person creating the grammar.

3. Finally, we formulate the grammar that describes the text fragment in the textbook. This can be done straightforwardly by reading off the grammar rules from the internal nodes in the trees (see Figure 9). This will usually result in an over-generating grammar, so we use different techniques to reduce the over-generation such as merging two or more generated rules into one.

```
Caesar Augustus
imperator Romanus
est.
```

Figure 7.: Example sentence

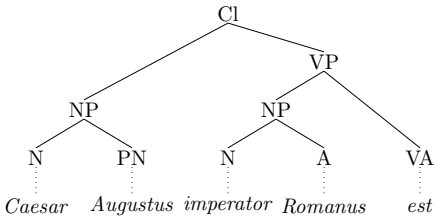


Figure 8.: Syntax tree derived from sentence in Figure 7

Abstract:

```
NP —> N PN
NP —> A N
VP —> VA NP
Cl —> NP VP
S  —> Cl
...
```

Figure 9.: Derived abstract grammar

3.4. Making the lesson grammars ungrammatical

The above process yields a grammar which only accepts syntactically correct sentences. However, we also wish to train the morphology in the object language, e.g. noun-verb agreement, number/gender inflection, affixation, etc.

It is possible to semi-automatically transform a lesson grammar into a grammar that accepts some grammatical errors, e.g. sentences where nouns and verbs disagree in number. What has to be done manually is to indicate which inflection parameter(s) in which grammar rule(s) should be loosened. Then it is possible to automatically transform the grammar into a grammar that accepts sentences where that specific inflection parameter is violated.

The user's task still is to edit a sentence in the object language to make it a translation of the meta language, but now they will have the additional complexity of allowing ungrammatical sentences. It is possible to control the level of ungrammaticality by deciding how many inflection parameters should be loosened.

3.5. Characterisation of the lesson grammars

We identified relevant criteria that characterise grammars that are suitable to be used by our application. The two most relevant criteria are a layer of semantics in the grammars as well as making some implicit features of the syntax explicit.

Grammars in GF usually are distinguished between resource grammars and application grammars. The main difference is that resource grammars just describe the syntax of a language without any semantic considerations while application grammars are used for a specific application and include semantic aspects necessary for that domain.

The grammars that are used in our system also have a strong focus on syntax, but require at least some semantic restrictions. From a purely syntactic point of view, adjectives can be combined with any noun, but language use only allows certain combinations. This can be solved by including semantic knowledge in different ways, of which the inclusion of FrameNet-style semantics in the grammars [19] seems the most natural.

The second point is, that natural languages might employ syntactic features that are not visible on the surface. One example are romance languages that allow the dropping of pronouns in the subject position because the relevant information is already present in the verb form. But to keep the grammars multilingual, these pronouns still have to be present in the grammar as empty tokens, which is an obstacle in language learning. This has to be changed in a way to make this information explicit on the surface for our application to be useful.

3.5.1. PENS classification

Given this description of the family of grammars we designed we want to render this definition more precisely in the PENS classification scheme [1]. PENS stands for Precision, Expressiveness, Naturalness, and Simplicity and is typically used to classify CNLs. Each of these scales ranges from 1 to 5 with 1 being the least and 5 the highest level of strictness possible.

Our grammars are fully specified in a computational grammar formalism and each sentence can be mapped to a set of abstract syntax trees. We do not insist on completely unambiguous interpretations but the set of interpretations will always be finite. According to the PENS classification that places our languages in the field of *Deterministically interpretable languages* (P^4).

The sentences generated by our grammars should be syntactically correct and in this aspect be considered as *Languages with natural sentences* (N^4). Because we focus just on sentences in our application, an extension to *Languages with natural texts* is not necessary.

The scope of our grammars is very limited, both by the text fragments and the explicit vocabulary, which makes it possible to formalise the language fragments in compact grammars. Even though they rely on external resources in the form of the RGL, the fragments can be considered as *Languages with short descriptions* (S^4).

The only problematic dimension is Expressivity, because we do not really focus on a translation to a specific logic interpretation, but remain on the level of the abstract syntax tree and its expressiveness. But because this is not relevant for our application we decide to ignore this dimension and set it to E^- That places our languages in the family of $P^4E^-N^4S^4$ languages.

This classification is not just some characterisation of the grammars we use in our system now and the languages defined by them, but instead a general requirement for all grammars and languages that can be used in our framework.

4. Evaluation

To test the acceptance of our application, as well as the desired change in learning outcome and learner motivation, we designed an empirical evaluation which we partially conducted as a pilot in connection with an introductory course for Latin at university level.

The full experiment consists of a prepared set of four lessons with a runtime of about four weeks. At the beginning, the students are asked to answer a questionnaire to control for aspects of the learner background and give some insight into the motivation at the beginning of the course. A simple timed placement test with eight exercises, four from each lesson, estimates the language skills before taking the class. The participants then are split into one treatment group and one or several control groups. In the following four weeks the students in the treatment group get access to lessons matching the progress in class while one control group only gets access to the traditional learning material in the text book. After the experiment period the students are given a slightly modified version of the questionnaire from the beginning to test for a change in motivation and a second placement test to see if there is a change in speed to solve the exercises.

In the pilot, due to lack of students, we could only ask for general feedback without gaining relevant insight into change in learner motivation or learning outcome. From ten students in class six volunteered to try the application and answer the first questionnaire. But due to a general drop out from this class, only four students were present in the end, of which only two had volunteered to participate. Still, the general feedback from both teachers and students was very positive which encourages us to aim for a full scale version of the evaluation.

5. Discussion

In the related work we pointed out several different technical approaches for systems in the field of foreign and second language learning. The different systems differ not only in the expressivity of the underlying technology but also in their intended use case.

Systems which employ technology with limited expressivity like finite-state technology aim at a closed setting in a very specific classroom setting but provide a high reliability. Other systems that employ very expressive machine learning methods can be used in a very open and classroom-independent setup but suffer from a lack of reliability.

With our system, which uses a very expressive syntax formalism, we currently target a closed classroom setting where we can profit from the reliability of our grammar-based approach but we also believe it is possible to widen the focus to provide a completely open language-learning application.

We claim that our system employs controlled natural languages for language learning. Some might disagree, and we admit that the PENS classification fails in the point of expressivity. But the application we sketch is grammar-agnostic, which means one can use almost any multilingual grammar to generate translation exercises from it. The grammars we used so far might not really be seen as

controlled languages because they are defined too implicitly, even though textbook lessons usually are created with clear concepts in mind. Still it can be used with any grammar that fulfils the PENS requirements we identified as characteristic for our grammars. This also gives a chance for further research looking into the application of CNLs in CALL far beyond the scope of this work.

Finally it is possible to discuss the combination of the underlying technology with other CNLs to build different applications. We think that there is some potential, especially given the similarity of conceptual editing and the word-based text editing, to have a fruitful exchange between the CNL community and other disciplines.

6. Conclusions and Future Work

We presented a working application usable for language learning that uses fully formalised grammars to define language learning lesson. According to some definition of controlled natural languages these lessons can be seen as CNLs, even there might be problems with this claim.

In the future we want to investigate how the design of the grammars influences the learning experience. This mostly concerns the structure of the grammars with varying focus on syntax and semantics. But that also includes additional ideas for different kinds of language learning exercises.

Another relevant topic of research is automatic generation of "good" exercises. This in entangled with the questions which kind of exercises besides translation exercises we want to include in our application. It also seems connected to a different topic, the selection of good examples in the creation of lexica [20], even though features of good translation exercises are not exactly the same as for good lexicon examples. Still this would give an opportunity for further interdisciplinary research.

References

[1] T. Kuhn, "A survey and classification of controlled natural languages," *Computational Linguistics*, vol. 40, pp. 121–170, March 2014.

[2] C. Hallett, D. Scott, and R. Power, "Composing questions through conceptual authoring," *Computational linguistics*, vol. 33, no. 1, pp. 105–133, 2007.

[3] E. Abolahrar, "Multilingual grammar-based language training: Computational methods and tools," Master's thesis, Chalmers University of Technology, 2011.

[4] T. Kuhn and R. Schwitter, "Writing support for controlled natural languages," in *Proceedings of the Australasian language technology association workshop 2008*, pp. 46–54, 2008.

[5] K. Angelov and A. Ranta, "Implementing controlled languages in GF," in *Proceedings of the 2009 Conference on Controlled Natural Language*, CNL'09, (Berlin, Heidelberg), pp. 82–101, Springer-Verlag, 2010.

[6] N. E. Fuchs, S. Höfler, K. Kaljurand, F. Rinaldi, and G. Schneider, "Attempto Controlled English: A Knowledge Representation Language Readable by Humans and Machines," in *Reasoning Web, First International Summer School 2005, Msida, Malta, July 25–29, 2005, Revised Lectures* (N. Eisinger and J. Małuszyński, eds.), no. 3564 in Lecture Notes in Computer Science, Springer, 2005.

[7] A. Ranta, "Grammatical Framework: A multilingual grammar formalism," *Language and Linguistics Compass*, vol. 3, no. 5, pp. 1242–1265, 2009.
[8] A. Ranta, *Grammatical Framework: Programming with Multilingual Grammars*. Stanford: CSLI Publications, 2011.
[9] C. K. Ogden, *Basic English : A general introduction with rules and grammar*. London: K. Paul, Trench, Trubner & co., ltd, 1930.
[10] H. Kaya and G. Eryiğit, "Using finite state transducers for helping foreign language learning," in *Proceedings of the 2nd Workshop on Natural Language Processing Techniques for Educational Applications*, pp. 94–98, 2015.
[11] M. Moritz, B. Pavlek, G. Franzini, and G. Crane, "Sentence shortening via morpho-syntactic annotated data in historical language learning," *Journal on Computing and Cultural Heritage (JOCCH)*, vol. 9, no. 1, p. 3, 2016.
[12] L. N. Michaud, "King Alfred: A translation environment for learners of Anglo-Saxon English," in *Proceedings of the Third Workshop on Innovative Use of NLP for Building Educational Applications*, (Columbus, Ohio), pp. 19–26, Association for Computational Linguistics, June 2008.
[13] H. Redkar, S. Singh, M. Somasundaram, D. Gorasia, M. Kulkarni, and P. Bhattacharyya, "Hindi shabdamitra: A WordNet based e-learning tool for language learning and teaching," in *Proceedings of the 4th Workshop on Natural Language Processing Techniques for Educational Applications (NLPTEA 2017)*, pp. 23–28, Asian Federation of Natural Language Processing, 2017.
[14] A. K. Horie, "Rewriting Duolingo's engine in Scala." `http://making.duolingo.com/rewriting-duolingos-engine-in-scala`, January 2017. accessed 04.04.2017.
[15] S. Ehrling, *Lingua Latina novo modo – En nybörjarbok i latin för universitetsbruk*. University of Gothenburg, 2015.
[16] P. Ljunglöf, "Editing syntax trees on the surface," in *Nodalida'11: 18th Nordic Conference of Computational Linguistics*, (Rīga, Latvia), 2011.
[17] G. Détrez and A. Ranta, "Smart paradigms and the predictability and complexity of inflectional morphology," in *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, EACL '12, (Stroudsburg, PA, USA), pp. 645–653, Association for Computational Linguistics, 2012.
[18] A. Ranta, "The GF Resource Grammar Library," *Linguistic Issues in Language Technology*, vol. 2, no. 2, 2009.
[19] N. Gruzitis and D. Dannélls, "A multilingual FrameNet-based grammar and lexicon for controlled natural language," *Language Resources and Evaluation*, vol. 51, pp. 37–66, Mar 2017.
[20] A. Kilgarriff, M. Husák, K. McAdam, M. Rundell, and P. Rychlý, "Gdex: Automatically finding good dictionary examples in a corpus," in *Proceedings of the 13th EURALEX International Congress* (E. Bernal and J. DeCesaris, eds.), (Barcelona, Spain), pp. 425–432, Institut Universitari de Linguistica Aplicada, Universitat Pompeu Fabra, jul 2008.