UNIVERSITÉ DE LORRAINE
IDMC

MASTER THESIS

Semi-automatic generation of English grammar exercises with rule-based and deep learning techniques

Author:
Chrysovalantis MASTORAS

Supervisor:
Prof. Yannick PARMENTIER

Jury:
Maxime AMBLARD
Philippe de GROOTE
Yannick PARMENTIER
Sylvain POGODALLA

A thesis submitted in fulfillment of the requirements
for the degree of Master of Natural Language Processing

in the

Synalp team
Loria

1 March 2021 - 31 July 2021

Declaration of Authorship

I, Chrysovalantis MASTORAS, declare that this thesis titled, "Semi-automatic generation of English grammar exercises with rule-based and deep learning techniques" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed: _____

Date: _____

UNIVERSITÉ DE LORRAINE

Abstract

IDMC
Loria

Master of Natural Language Processing

Semi-automatic generation of English grammar exercises with rule-based and deep learning techniques

by Chrysovalantis MASTORAS

During the last recent years, there has been reported an augmenting incorporation of Natural Language Processing (NLP) techniques on various educational settings. Such approaches usually leverage some of the state-of-the-art advances of the NLP field, to enhance the language learning procedure by reducing the human intervention on various tasks.

Following this research trend, we have developed a framework that can generate Second Language (L2) exercises with NLP methods, in a semi-automatic manner, namely *SemiGramEx*. More precisely, given a set of input parameters by a user, the framework can generate a set of grammar exercises. While most of the reported approaches on such a subject, in our knowledge, had been targeting the aid of L2 learners, in this work the target user is an L2 teacher. More specifically, this framework aims on diminishing some of the most time-consuming tasks that a L2 teacher might have to confront, during the preparation of a L2 teaching setting (such as the manual creation of appropriate L2 exercises of that sort). However, this work does not aim on eliminating the participation of the teacher. On the contrary, SemiGramEx is presented as a *semi-automatic generation system*, since the teacher is considered to be an actively participating component of the generation process, who must revise and reform, when necessary, the framework's generation results.

In more details, SemiGramEx accepts a set of input parameters, such as a target teaching goal and a target difficulty level, and it generates a set of L2 grammar exercise instances. The framework supports for the moment verb tenses learning, with 6 verb tenses being employed, 3 different types of exercises and 3 difficulty levels. A combination of rule-based and deep learning techniques has been leveraged for all these to happen and a functional web-interface has also been implemented.

The results that are reported on various evaluation dimensions, are already indicating that the framework is in an adequate state for the task in hand, even though there is still some room for improvement.

Acknowledgements

At first, I would like to express my deepest gratitude to my supervisor, Yannick Parmentier, without whom this work would not be possible to be concluded. I am particularly grateful for all the time he has spent on me, the excellent working environment and communication, as well as the instant support on anything that had occurred during the internship period. All these details had rendered this internship a very pleasant experience.

In addition, a special gratitude to all the professors that have participated during the two years of this Master degree, as well as my student colleagues for all the work, the discussions, and the perspectives we have shared during these two years. All these stimuli have been greatly motivating for my future evolvement.

Finally, I would like to thank the Loria research center and the Synalp team, for hosting my internship as well as the University of Lorraine and the IDMC.

Contents

List of Figures

List of Tables

List of Abbreviations

biLSTM	**bi**directional **L**ong **S**hort-**T**erm **M**emory
BNC	**B**ritish **N**ational **C**orpus
CALL	**C**omputer **A**ssisted **L**anguage **L**earning
CEFR	**C**ommon **E**uropean **F**ramework of **R**eference for Languages
DL	**D**eep **L**earning
FIB	**F**ill-**i**n-the-**b**lank
HNN	**H**ierarchical **A**ttention **N**etwork
IAA	**I**nter-**A**nnotator **A**greement
ICALL	**I**ntelligent **C**omputer **A**ssisted **L**anguage **L**earning
L1	**F**irst **L**anguage
L2	**S**econd **L**anguage
LM	**L**anguage **M**odel
ML	**M**achine **L**earning
NL	**N**atural **L**anguage
NLP	**N**atural **L**anguage **P**rocessing
RNN	**R**eccurent **N**eural **N**etwork
ScoRE	**S**entence **c**orpus **o**f **R**emedial **E**nglish
SLA	**S**econd **L**anguage **A**cquisition
SVM	**S**upport **V**ector **M**achines
UD	**U**niversal **D**ependencies

Chapter 1

Introduction

1.1 Project overview

Since the 1960s many research efforts have been reported, that tried to integrate the scientific domain of NLP with education. More specifically, Computer Assisted Language Learning (CALL) and more recently, Intelligent Computer Assisted Language Learning (ICALL) are two research domains upon which NLP applications in an educational context have flourished over the last recent years. Those two domains aim on helping language learning by reducing human intervention with the aid of automatic methods and therefore, they can be seen as complementary if not alternative solutions to conventional language learning practices.

Under that light, and due to the dramatic growth of easily accessible Natural Language (NL) text data on the Web, one could aim on incorporating NLP techniques and reclaim such available NL data, to create appropriate educational content that might facilitate a language learning process. For such a purpose, some of the most typical and well-known NLP modules might prove to be valuable allies. Two famous NLP modules that interest us the most in this work are those of *NL parsing* and *Deep Learning (DL)*, where the former is an NLP technique that is used to build a representation of the internal structure of a text, while the latter is a recent, state-of-the-art modeling architecture that has shown exquisite results on a diversity of complex tasks. As a result, the aim of our work is to present a concrete framework where those two modules, along with a diversity of other NLP techniques, will be simultaneously employed to produce valuable educational resources from a collection of available NL text data. More specifically, in this work, the target is L2 learning and therefore, what we will present is the implementation of a generation system for L2 exercises, that combines DL and rule-based methods to enhance, if possible, the learning procedure. We have named that generation system *SemiGramEx*, and we will continue referring to it in that way for the following part of our work.

While a diverse set of possible exercises or activities exist for such L2 learning settings, such as vocabulary acquisition and oral/written expression, the main target of this work is conjugation acquisition. Such a teaching goal is usually accomplished with the practice and evaluation of an L2 learner on typical grammar exercises (i.e., fill-in-the-blank exercises). Therefore, our aim is to present a solid generation framework capable of instantly creating such grammar exercises. More precisely, the general teaching goal that our system supports for the time being is *verb tenses learning*, a grammatical goal that is most widely used in any L2 learning setup, and the target language being chosen is English, while plans to extent this work on other languages also exist. In addition, even though most of the existing research on that subject has been targeting L2 learners, our work is focused on the aid of L2 teachers with the automatic construction of the educational content to be taught of, a task that is quite

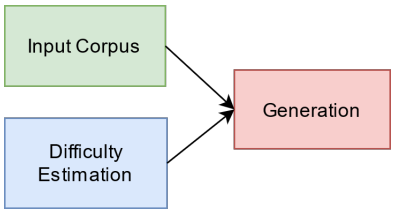


FIGURE 1.1: The three modules of SemiGramEx.

frequent on the duties of any L2 teacher. More specifically, given a set of input parameters by the teacher, the system's goal is to instantly generate concrete grammar exercises that target the teaching of verb tenses. Those exercises, might be used directly or after a reform by the teacher, in a L2 learning setup. In that sense, the generation framework in hand is a semi-automatic one, since the L2 teacher can be considered to be an active and indispensable component of the generation procedure, who must revise the generated results before demonstrating them to a target learner.

Eventually, concerning the details of the implementation, SemiGramEx is consisted of three major modules. An *Input Corpus* module, a *Difficulty Estimation* module and eventually a *Generation* module. The three modules of the pipeline can be seen on Figure 1.1. For the Input Corpus module, a collection of written texts has been retrieved and processed in an appropriate manner for the generation task in hand. That collection was further annotated with different layers of linguistic information that are relevant to the generation procedure and its goal is to constitute the generation basis of SemiGramEx. Additionally, the role of the Difficulty Estimation module is to categorize the text material collected on the Input Corpus module into appropriate language proficiency levels, a task that is necessary for any L2 learning setup. For that reason, a DL classifier has been incorporated where given an input sentence, a proficiency level of that sentence can be estimated. Eventually, the Generation module is the component that joins together the first two modules of our pipeline and generates the final grammar exercises. More precisely, given a target teaching goal, the system retrieves appropriate candidate sentences from the text collection created on the Input Corpus module. Those candidate sentences are further labeled with a difficulty scale, based on the predictions of the Difficulty Estimation module and, given an additional parameter of a difficulty level by the teacher, they are filtered correspondingly. Then, those final retrieved sentences are being employed by the Generation module, where they are converted into appropriate grammar exercises with the use of rule-based criteria and hand-crafted transformation patterns. SemiGramEx supports for the moment three different types of grammar exercises which will be subsequently elaborated and all the aforementioned components constitute a complete generation framework with a web-based interface that is ready to be used and tested.

To conclude this introduction, the general research question with which this thesis is being occupied, can be stated as:

- Is it possible to use NLP techniques to help L2 teachers automate the tedious task of grammar exercises construction?

To explore that question, it is necessary to further analyze it into two sub-parts:

1. How to retrieve from a given text passage, a set of sentences that are adequate for a target L2 learner? (In terms of a target teaching goal and a language proficiency level)

2. How to convert such extracted sentences into appropriate L2 grammar exercises?

The rest of this thesis is structured as follows. In the next section of this Chapter, we will shortly present the working environment where the internship and the thesis were conducted, along with the difficulties that were emerged during our work. Then, on Chapter 2 we will introduce the first module of our pipeline, the Input Corpus, where an overview of relevant examples from the recent literature will be presented, as well as the approach that is being leveraged in this work. Subsequently, the Difficulty Estimation module will be presented on Chapter 3. An overview over the long history of such difficulty estimation tasks will be reported, as well as some of the recently presented state-of-the-art approaches. In addition, the approach that is adopted in this work to tackle the difficulty estimation problem, along with the implementation details of it, will also be reported on the same Chapter. Moreover, Chapter 4 will be devoted on existing grammar generation approaches that have been presented over the past recent years, as well as the generation approach that is followed on our Generation module. Eventually, on Chapter 5, a short evaluation that was conducted to obtain insights about the usefulness of our approach will be presented, while on Chapter 6, the conclusion of this work will be reported and we will also discuss about possible improvements of our framework.

1.2 Working environment

The work being described in this thesis was conducted as part of a last year Master degree internship, under the supervision of professor Yannick Parmentier. More specifically, this work is in the frame of a bigger research project, dubbed *GramEx* (Grammar Exerciser). GramEx is part of a prototype project, namely the METAL[1] project, that is funded by the French Minister of Education and its goal is to develop tools for learners and teachers. It is hosted by the Synalp[2] research team at the LORIA[3] research center. Following in this section, we will thoroughly present more details about the working environment of this internship, the tools that were utilized, as well as the difficulties that were emerged during it.

1.2.1 LORIA and the Synalp team

LORIA is a French research unit (UMR 7503) that was created in 1997. Its name is a French acronym for *Lorraine Research Laboratory in Computer Science and its Applications*. The lab is directed by Jean Yves Marion and a direction team, along with a scientific council, a lab council, and the responsible researchers of each research team. It is shared between three institutions, those being the *French National Center for Scientific Research* (Centre National de la Recherche Scientifique, CNRS), the *University of Lorraine* (Université de Lorraine) and the *National Institute for Research in Digital Science and Technology* (INRIA). The lab's goal is to deal both with the fundamental, as well as the applied research in computer sciences.

Around 400 people are working at the LORIA lab and 29 research teams are composing it. LORIA is structured into five departments and each one of the 29 research teams belongs to one of them, depending on the research domain of interest. More

[1] http://metal.loria.fr
[2] https://synalp.loria.fr/
[3] https://www.loria.fr/en/

specifically, the existing research departments at LORIA are the *Algorithms, Computation, Image and Geometry* department, which focuses on geometry and symbolic computations, the *Formal Methods* department, which focuses on software-based systems that incorporate formal methods, the *Networks, Systems and Services* department, which focuses on computer networks, as well as parallel and distributed systems, the *Natural Language Processing and Knowledge Discovery* department, which is oriented on natural language and knowledge modeling, and the *Complex Systems, Artificial Intelligence and Robotics* department, which focuses on artificial intelligence and robotics. In addition, LORIA is actively involved on many French and international industrial collaborations.

The Synalp team, is one of the 29 research teams of LORIA. It is a part of the *Natural Language Processing and Knowledge Discovery* department, and its research interests are focused on hybrid, symbolic and statistical natural language processing approaches. Its research domains include Language Models, Formal Grammars, Natural Language Processing, Computational Semantics and Speech Processing, where some current research topics among others concern Natural Language Generation, Human-Machine Dialog, Text-to-speech Alignment, Language Learning etc. The head of the Synalp team is Christophe Cerisara and the supervisor of this work, Yannick Parmentier, is a researcher of it.

1.2.2 Work setup and emerged difficulties

Concerning the internship, that was almost exclusively conducted remotely, due to the health constraints that were imposed by the COVID-19 pandemic. The only equipment that was used during our implementation, were a personal laptop and a desktop personal computer. The main programming language that was chosen was Python[4], along with the Anaconda environment[5]. The main DL framework that was incorporated was PyTorch[6] and various of the major data science libraries were employed (such as NLTK[7], ScikitLearn[8], Pandas[9], Spacy[10], etc.). In addition, a web-interface of the overall implementation was created. For that reason, the Flask[11] framework was incorporated, as well as the HTML5, CSS3 and JQuery[12] programming languages. Google Colab[13] was leveraged as the main coding environment, even though the Jupyter Notebook[14] interface and the PyCharm[15] IDE, were also employed in some extent, mostly for tasks that were mandatory to be implemented locally. More specifically, Google Colab was selected as the main implementation environment for two reasons. At first, it offers a free access on available graphical processing units, a very important resource for any computationally intensive task and secondly, Google Colab is a much easier setup when it comes to the installation of various libraries and their dependencies, since it can simplify time-consuming

[4] https://www.python.org/
[5] https://www.anaconda.com/
[6] https://pytorch.org/
[7] https://www.nltk.org/
[8] https://scikit-learn.org/
[9] https://pandas.pydata.org/
[10] https://spacy.io/
[11] https://flask.palletsprojects.com/en/2.0.x/
[12] https://jquery.com/
[13] https://colab.research.google.com/
[14] https://jupyter.org/
[15] https://www.jetbrains.com/pycharm/

procedures of that kind in a great extent. During the production stage of our implementation, a Google Drive[16] workspace was created where each one of our implementation modules were separated into different working directories (each directory contained all the necessary files of the corresponding implementation module). In that way, an efficient and remote coding environment was created on the cloud, that could also be easily shared with the supervisor of this project. Furthermore, a direct interface between Google Colab and Google Drive exists. In that way, all of the implementation code could always be accessible on the cloud, and it could also be easily processed with the computational resources offered from Google Colab, a fact that intensively enhanced the implementation procedure. On the contrary, for the development and testing stage of our implementation, a Github[17] repository was created where the code of this work is shared, and the reader might also have the chance to examine many details of it, in the corresponding README.md file.

Conclusively, during this internship period, a set of difficulties were emerged which should be mentioned. More specifically, the lack of continuous and instant aid due to the remote nature of this internship and the absence of colleagues, appeared to be rather challenging. Even though the frequent online meetings with the supervisor of the internship had greatly improved that situation, it was still challenging to operate in such a setting. However, all those difficulties appeared to be very fruitful eventually, since qualities such as self-organization and autonomicity, were rapidly developed. In addition, another challenge that was confronted, had to do with the nature of the research project in hand. More precisely, as it was previously mentioned, this work is only an initial part of a bigger, long-term project. Therefore, many different aspects of it are still vague and under discussion/research. As a result, instead of beginning this work directly with a strict and precise research goal, an extensive research and many experiments had to be conducted initially, so that a feasible research topic proposal would be built. That task was perhaps the most challenging one that had to be confronted, mainly due to the inadequate experience on such procedures. Nevertheless, that challenge was also the most educative one during this internship period, since it gave the chance of confronting a real-word research example from the very first step to the very last of it, rendering this internship period a holistic research experience.

[16] https://www.google.com/drive/

[17] https://github.com/Valadis-Mastoras/SemiGramEx

Chapter 2

The Input Corpus module

The first component of our work is the *Input Corpus* module that was also previously mentioned. During that part of the implementation, the first thing we had to decide was the generation approach that our framework should follow. As it appears, two main approaches exist on the relevant research literature for the task of automatic grammar exercises generation. The first approach is to incorporate a dedicated input resource as a generation basis (indicatively, Lee and Luo, 2016) and the other is to directly generate exercises with the aid of automatic text generation techniques (indicatively, Perez-Beltrachini, Gardent, and Kruszewski, 2012). Even though both approaches are present on the recent literature, most of the proposed generation approaches for the task of automatic generation of grammar exercises, in our knowledge, had incorporated a dedicated input resource as a basis of the generation procedure. As a result, we too have chosen to follow a similar path due to the validity it was demonstrated in the literature.

Following in this Chapter, we will initially present some of the most frequent classes of input resources that have been used in the relevant literature on similar tasks. We will be focused on the necessary qualities of such input resources, as well as the aspects that render them suitable for a language learning setting. In addition, we will thoroughly report all the decisions and the steps that were made during the implementation of the Input Corpus module. Moreover, we will continue this Chapter under the assumption, that the incorporation of an input resource as a generation basis for the task of automatic grammar exercises generation, is indeed a valid approach. However, in the subsequent Chapter 4, some of the most indicative generation methods on the relevant literature will also be reported, where the reader might then have the chance to examine the research trends of the last recent years and verify the validity of our chosen generation approach.

2.1 Input resources used in the literature

Having decided to incorporate a dedicated input resource as a generation basis, the next decision we were called to make concerned the specific qualities that such a resource should carry. Even though an abundance of possible text resources exists on the Web, it was necessary that the incorporated input resource would be suitable for the specific nature of our task. For that reason, we had intensively studied on the relevant research literature for sources of inspiration.

Three main classes of input resources appear to have been widely used in the literature for the task of automatic generation of language exercises. The first class of resources is the incorporation of *second language textbooks*, as in the case of Soonklang and Muangon, 2017, where pedagogically oriented and well-formed text material can be obtained before being converted into the appropriate exercise form.

Subsequently, the second resource class, which was mostly reported on the early years of the relevant research, was the incorporation of an *input material retrieved from the Web*. Those are for example the cases of Mitkov et al., 2003 and Chen, Liou, and Chang, 2006, who based their generation on documents and sentences retrieved from the Web, or Hoshino and Nakagawa, 2005 and Marrese-Taylor et al., 2018, who leveraged data from an online language learning platform and on-line news articles, accordingly. Eventually, the third class of widely used input resources, is the generation from *pre-existing collections of corpora*. Indicatively, Fenogenova and Kuzmenko, 2016 had used as a generation basis two major corpora namely the *British Academic Written English Corpus* (Alsop and Nesi, 2009) and the *British National Corpus* (Leech, 1992), where the former is an English corpus of academic written texts covering a broad range of discipline areas and difficulty levels, while the latter is a collection of texts of written and spoken British English for various topics of the late twentieth century and different ages. Additionally, other input generation resources that were proposed, might be the *Project Gutenberg*[1], a collection of 57,000 freely available e-books in 67 languages, or documents collected from the *Wikipedia*[2] and the *Simple Wikipedia*[3] database (the cases of Agirrezabal et al., 2018; Lee and Luo, 2016; Heilman and Smith, 2010, accordingly). Eventually, even though much less frequently, parallel corpora have also been incorporated, with a specific language being selected as the target text resource, as in the example of Chalvin, Eensoo, and Stuck, 2013.

As it was illustrated, a great diversity of leveraged input resources have been reported on the relevant literature for generation tasks like ours. Those input resources might be retrieved from sources that are specific to language learning, as for example dedicated L2 textbooks, but they can also be as free and open class as resources obtained directly through the Web. Therefore, the language being used in such resources might be intensively balancing between pedagogical appropriate and pedagogical inappropriate use of language. That raises a major dilemma, concerning the accepted form of language that an input resource for L2 learning should have.

Two main types of language examples for L2 exercises might be distinguished. Those are the *synthetic language examples* and the *authentic language examples*, where the former is a form of language that is tightly conditioned in terms of pedagogical validity, and the latter is an example of real-world language usage. To elaborate more on those two categories, the synthetic language examples have been in fact a widely used case of language learning material. These are constructed language examples that come with the advantage of being controllable and therefore specific to a target learner, but at the same time they are quite ideal language constructions which are not the actual norm in a spoken language. In addition, such a way of creating educational content is tightly depended on a group of experts, without whom the creation of such exercises is impossible. On the other hand, the incorporation of authentic language examples usually comes without any manual effort, since such examples might be easily retrieved from every-day resources, such as the abundance of free text that exist on the Web. That sort of language resources come also with the advantage of being authentic, meaning that it is the actual kind of spoken language that a learner would have to encounter in a native speaker environment. Nevertheless, that kind of input material might contain many infrequent words and more importantly, the very notion of authentic language use might have a rather different

[1] https://www.gutenberg.org/
[2] https://en.wikipedia.org/wiki/Main_Page/
[3] https://simple.wikipedia.org/wiki/Main_Page/

meaning for First Language (L1) and L2 learners. In that sense, it might be appropriate to leverage such a language use when dealing with a L1 learner, and at the same time such language use examples might be inappropriate for a L2 learner.

In this work, we have decided to incorporate an input resource that will be consisted of authentic language use examples. The main reason to do that, had to do with the way we have chosen to view the generation task in hand and in extent, the collection procedure of our input resource. More specifically, we have decided to treat the input resource collection procedure in a computationally oriented way, rather than a pedagogically oriented one. What that means, is that we are primarily concerned on how and whether we could incorporate state-of-the-art NLP techniques to transform an abundance of freely available reading material retrieved by the Web, into useful educational material for L2 teachers. Even though the pedagogical validity of that reading material might be an aspect of great importance, it has only consisted of a secondary concern for us during the collection procedure of the input resource. (However, it should be mentioned that this perspective concerns only the collection procedure of the input resource, meaning that we were interested on whether we could obtain remarkable generation results without the necessity of constructing the input resource under specific pedagogical guidelines. For the final generation outcomes, as well as the overall quality of our framework, pedagogical validity maintains the most important role). That decision, was mostly taken since our generation scenario is not directly targeting L2 learners but L2 teachers, who will play the important intermediate role of a proofreader. Due to that fact, many degrees of freedom were acquired in our work since even if possible discrepancies, in terms of pedagogical validity, might be inherited on the generated exercises from the input resource in use, the teacher might always have the chance to evaluate the results and possibly reform them in an appropriate manner. Furthermore, the research of Hubbard, 2012 had also empowered our position at that point. In this work, it was argued that authentic language material can be proper for a language learning setting if the approach is not entirely automatic, but rather a human proofreading will also be part of the generation procedure, before the results are demonstrated to a learner. Deriving from all that, we were greatly optimistic that the use of authentic text would be suitable for an L2 learning setting and therefore, that was the path we had followed.

2.2 Input resources used on SemiGramEx

To implement the Input Corpus module, three additional sub-components were developed, with those being the *Scraping* component, the *Parsing* component and the *Sentence Selection* component. At first, the Scraping component was responsible of scraping and pre-processing a collection of target texts that would constitute the input resource in use. Subsequently, on the Parsing component, different layers of linguistic annotations were added to the collected input resource, to enrich it. Eventually, on the Sentence Selection component, an extra filtering of the collected input resource was made, so that only the most meaningful and less ambiguous parts of it, would remain. An illustration of all those components is demonstrated on Figure 2.1 and each one of them, will be thoroughly presented subsequently.

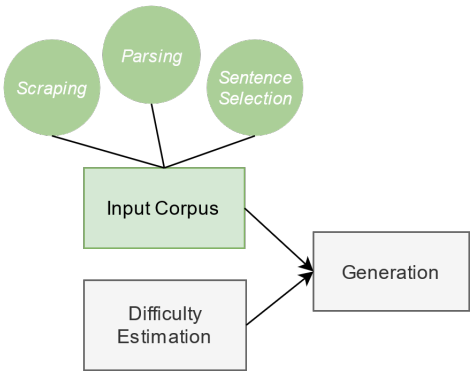


FIGURE 2.1: The components of the Input Corpus module.

2.2.1 The Scraping component

The first component of the Input Corpus, namely Scraping, was responsible of collecting an appropriate text material and converting it into the appropriate form. Having decided that we will incorporate an input text resource of authentic language as a generation basis for our system, we began to thoroughly search for possible resources of that kind. The input resource research had been quite an extensive one and many different resources were examined for their credibility in our task. We conducted our research under the conditions that the input resource should be freely available, easily accessible, adequate in terms of size length and that it would demonstrate an authentic language use. In addition, it had to retain in a minimum extent some formal language guidelines that would restrict examples of ungrammatical use of language or informal use of language, such as the case of the slung language.

Among many different candidates, the Wikipedia database appeared to be one of the most tempting resources for our purpose. That database does not only contain an abundance of freely accessible text data, but it also comes with the advantage of being created under very specific writing guidelines that every article entry should retain. However, the writing style that is usually adopted on Wikipedia articles might be a little formal for a L2 learning setting. Therefore, we decided to incorporate the alternative, Simple Wikipedia resource, as the most prominent input resource candidate for our task. While the Wikipedia database is oriented mostly to adults with an adequate language level, the Simple Wikipedia version of Wikipedia was constructed specifically for readers with a less solid language proficiency level, as it was also previously mentioned. More specifically, the general guideline that is given to the Simple Wikipedia content writers, is to maintain a style of writing that is dedicated to adolescents or L2 learners. As a result, Simple Wikipedia was considered as a valid resource that sufficiently fulfilled all the necessary conditions we had set.

A collection of 10,000 articles were scraped from the Simple Wikipedia database. Those articles were retrieved with the Wikipedia API[4] based on the list of *vital articles of level 4*[5] that Wikipedia offers. The Wikipedia vital articles are mainly article lists that are published by Wikipedia, and concern a diverse range of subjects that should exist in a Wikipedia database. After being scraped, those articles were pre-processed and cleaned from any text noise, to maintain only the main content of each article and the most relevant information. All the pre-processing steps were

[4]https://pypi.org/project/wikipedia/
[5]https://en.wikipedia.org/wiki/Wikipedia:Vital_articles/Level/4

designed specifically for the structure of a Simple Wikipedia article and eventually, each of those clean scraped articles were split into a collection of sentences. Having performed all these steps, we ended up with a collection of properly processed sentences, retrieved through the Simple Wikipedia database.

In addition, we incorporated a second input resource, so that SemiGramEx would demonstrate the best possible variability on the generated exercises, in terms of topics being covered as well as the style of writing. For that reason, a subset of the British National Corpus (BNC) was also incorporated. As it was previously mentioned, the BNC is a collection of texts that covers a great diversity of topics and contexts, rendering it an excellent alternative candidate to the Simple Wikipedia resource. Nevertheless, the BNC corpus is only provided as an experimental input resource option for the moment, since some possible discrepancies may exist there, due to time limits for a more adequate processing. A more extensive research should be made in the future, on ways that would render it as adequate as possible for L2 exercises.

To create that secondary input resource, a BNC subset of 4M words namely the *small BNC corpus*[6] was incorporated. In a similar manner with the one that was described for the Simple Wikipedia corpus, several pre-processing steps were performed, which were precisely designed for the text form of the small BNC corpus. After those steps were executed, we eventually managed to acquire a second collection of appropriate sentences that were retrieved from the BNC corpus.

2.2.2 The Parsing component

Having acquired those two sentence collections, the next step was to retrieve additional linguistic information for those sentences. In that way, a set of queries could be performed on the input resource collections, to isolate specific sentences given a set of target linguistic constructions. For that reason, we incorporated the advances of a state-of-the-art parser of that kind. Such parsers can automatically annotate a given text passage with different aspects of linguistic information, such as the part-of-speech tags or other syntactical and morphological features of a sentence. While many possible options of such parsers exist, we leveraged the *Stanza parser* (Qi et al., 2020) as the most suitable candidate.

The Stanza parser is an open-source NLP tool that supports 66 human languages. It features a fully neural pipeline that retrieves raw text as input and produces various annotations such as the lemma form, the part-of-speech tags, the syntactic dependencies, or the morphological features. It was only recently published and has already demonstrated state-of-the-art results on various tasks. For the syntactic dependency relations, it follows the typical Universal Dependencies (UD) formalism. The main advantages, compared to existing parsing toolkits, is its fast-processing power, as well as the neural and language-agnostic nature of the parsing procedure, a fact that was of the uttermost importance for us, since future plans exist to expand this system on supporting multi-lingual settings as well.

The parsing procedure was similar for both the input resources. Each sentence was separately parsed with the Stanza parser and a set of linguistic annotations was retrieved for each of them. More specifically, the tokens of each sentence, along with the part-of-speech tags and the lemma form of each word in each sentence, were retrieved. In addition, their corresponding syntactic dependency relations were also retrieved. Moreover, a positional identifier for each token was acquired, as well as

[6] https://ota.bodleian.ox.ac.uk/repository/xmlui/handle/20.500.12024/2553

Sentence	Tokens	Lemma	Upos	Xpos
History is the study of past events.	[History, is, the, study, of, past, events, .]	[history, be, the, study, of, past, event, .]	[NOUN, AUX, DET, NOUN, ADP, ADJ, NOUN, PUNCT]	[NN, VBZ, DT, NN, IN, JJ, NNS, .]
A person who studies history is called a histo...	[A, person, who, studies, history, is, called,...	[a, person, who, study, history, be, call, a, ...	[DET, NOUN, PRON, VERB, NOUN, AUX, VERB, DET, ...	[DT, NN, WP, VBZ, NN, VBZ, VBN, DT, NN, .]

FIGURE 2.2: Part 1: The Simple Wikipedia dataframe after being parsed with the corresponding linguistic information.

Dependency	Features	id	Head
[(study, nsubj), (study, cop), (study, det), (...	[Number=Sing, Mood=Ind|Number=Sing|Person=3|Te...	[1, 2, 3, 4, 5, 6, 7, 8]	[4, 4, 4, 0, 7, 7, 4, 4]
[(person, det), (called, nsubj:pass), (studies...	[Definite=Ind|PronType=Art, Number=Sing, PronT...	[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]	[2, 7, 4, 2, 4, 7, 0, 9, 7, 7]

FIGURE 2.3: Part 2: The Simple Wikipedia dataframe after being parsed with the corresponding linguistic information.

the head token relations (each token in a sentence is the head of another token based on their syntactic dependency relations). Eventually, a set of morphological features were further acquired for each sentence, such as the Person number or the Voice of the verb. All that information enriched the two input resources and eventually, one dataframe containing all that knowledge for each resource, was created. In the end, the Simple Wikipedia corpus consisted of 195,891 individual sentences, while the small BNC corpus consisted of 197,208 sentences. In Figure 2.2 and Figure 2.3, an instance of the Simple Wikipedia dataframe is demonstrated.

All the aforementioned steps were executed to convert the two input resources into a proper form, so that they could be employed on the next steps of our pipeline. The pre-processing steps that were performed, had cleaned the sentences from any irrelevant information, while the parsing steps had revealed the hidden linguistic information. In that way, the system had arrived at a point where we could retrieve appropriate subsets of sentences, based on a set of target linguistic constructions, and those sentence subsets could then be converted into the appropriate final exercise forms.

Nevertheless, those sentences were still far from being adequate for our task. Given that they had been forcefully separated from their surrounding context, it appeared that a great extent of them were quite ambiguous and insufficient in terms of self-containment, rendering them inappropriate of being standalone L2 exercise instances. As a result, an extra step was mandatory to be employed, so that only sentences that are context-independent would be retrieved. For that purpose, we partially followed the work reported by Pilán, Volodina, and Borin, 2017, to apply a set of better sentence selection rules and filter out the two corpora into the most meaningful and less ambiguous parts of them.

2.2.3 The Sentence Selection component

In Pilán, Volodina, and Borin, 2017, a sentence selection framework was presented, where a set of extensive rule-based criteria were incorporated, so that only well-formed and context-independent sentences could be retrieved from a given corpus. Their intention was to create a framework that would be pedagogically-aware and therefore, their work was oriented specifically on L2 settings. The proposed framework was quite compact and it consisted many different selection criteria, concerning both lexical and structural aspects of a text.

Following that example, we also tried to employ similar selection criteria for our task. However, the target language being incorporated in the work of Pilán,

Volodina, and Borin, 2017 was Swedish, a fact that imposed many restrictions in our work. As a result, some of the proposed selection criteria were not able to be reproduced, either because of inadequacies on necessary English resources, or due to general differences between the Swedish and the English language. Therefore, we only incorporated a limited subset of those selection criteria that were considered as the most suitable for our task. The adopted criteria mainly belonged to the *Well-formedness* and *Context-independence* reported classes, leaving an open space for a future elaboration on more of the proposed categories.

More precisely, the first criterion that we applied, aimed to examine that every sentence would start with an uppercase letter and end with a strong punctuation (a dot, an exclamation mark, or a question mark). Such orthographic clues could eliminate any possible inconsistency caused by the parser during the sentence splitting process (meaning, sentences that were separated as such by the parser, while they should not). In addition, the *structural well-formedness* of the sentences was examined with the presence of a root and an ellipsis dependency. That means that every sentence should have a root dependency relation, a subject dependency relation and a finite verb, so that it would be considered as syntactically robust. Additionally, a criterion to select sentences in a range between 4 and 25 tokens was enforced, to maintain a sentence length that would be feasible for an L2 setting. All those criteria aimed on acquiring sentences that would be properly well-formed.

At first, several criteria were further leveraged to cope with *context-dependence*. The syntactical aspect of context-dependence was examined based on various *structural connectives* (i.e., *than*). However, the connectives that were leveraged, were manually retrieved from various linguistic resources, and thus they cannot be considered as exhaustive. With the incorporation of such structural connectives, one can identify paradigms of coordinating or subordinating conjunction, where context-depended clauses can appear as standalone sentences. Therefore, a selection criterion was applied to eliminate such cases. Based on that criterion, a sentence was considered as context-depended if it would start with a structural connective, unless that sentence would contain more than one clause or if that structural connective was a paired conjunction (i.e., *either...or*). In addition, a sentence selection criterion to resolve anaphoric expressions, which are expressions that typically refer to previously mentioned information in a given context, was also incorporated. Similarly with Pilán, Volodina, and Borin, 2017, we were focused on pronominal pronoun anaphora and third person singular pronouns mentions, as well as demonstrative pronoun anaphora. The non-anaphoric use of third person singular pronoun was ignored only if it was a pleonastic one, and pronouns that were followed by a relative clause introduced by *which*, were also ignored as well.

All the aforementioned criteria, were applied on both the Simple Wikipedia and the BNC corpora. As a result, the Sentence Selection component managed to further filter the sentences of our corpora into 150,222 sentences for the Simple Wikipedia Corpus and 114,900 sentences for the BNC corpus. Having those two sets of sentences ready, the first component of our pipeline was completed and those two input resources would be subsequently elaborated on the next modules of the generation procedure.

Chapter 3

The Difficulty Estimation module

Having our input resources cleaned and converted into a proper form for the generation task in hand, the next component of our pipeline is the *Difficulty Estimation* module. That module is responsible of classifying subsets of the collected input resources, into appropriate difficulty levels. Such a difficulty classification process maintains a decisive role on the exercises' preparation task for any L2 teacher, since it is necessary for a teaching procedure to be meaningful, that the provided educational texts will be in accordance with the target learner's proficiency level (as O'Connor et al., 2002 had also indicated). As a result, an exercise generation system like the one we are presenting in this work, would perform in a rather restrictive way without the presence of such an automatic difficulty classification module.

Typically, the task of difficulty classification of L2 educational content might be seen under different perspectives. Indicatively, the type of a given exercise, the difficulty of the linguistic constructions that exist in an exercise or even the use of the plain language itself, could be seen as decisive factors on a difficulty classification procedure. Moreover, different difficulty scales could be leveraged, such as custom or formally defined ones. The Common European Framework of Reference for Languages (CEFR) (Modern Languages Division, 2001) grading might be one of the most famous examples of such a formally-defined difficulty scale, where a range of learner proficiency levels along with a set of pedagogical guidelines, concerning the prerequisite knowledge that a learner should acquire in each of those levels, has been presented.

Concerning our task, we have decided to incorporate an approach that is oriented directly on the difficulty of the plain language in use, rather than any other difficulty aspect. In addition, a custom 3-level difficulty range was chosen, instead of a formally-defined one, with Beginner, Intermediate and Advance, being the difficulty labels. One important factor that dictated us to follow that path, similarly with the decisions we had made during the Input Corpus module implementation, was the absence of domain-experts that could guide us on selecting pedagogically valid difficulty measures. Moreover, that decision was also taken since we estimated that the plain use, of language can comprise some of the most determinant difficulty aspects in a language learning procedure. After all, the ability of a learner to comprehend a target exercise, should primarily start at the level of the language in use before further proceeding into the deeper pedagogical conditions of it. As a result, we were led into the subject of *Readability Assessment*, a field with an extensive research history for subjects of that sort, based on which we have decided to confront our task.

For the following part of this Chapter, we will thoroughly introduce the evolution of that field, starting from traditional approaches to more recent ones. In addition, we will indicate some of the most important perspectives, based on which a readability assessment task could be seen. The reason to perform such an extensive

overview, is to render the reader aware of the general spectrum of the field, so that all the fuzzy points, the complexity, as well as the main limitations that we had to confront during the implementation of it, might be clearly understood. Eventually, all the implementation details of the approach that we have followed, will also be presented.

3.1 Relevant literature

3.1.1 Overview of readability assessment

Readability assessment is a subject with long history that firstly originates back to 1920s and aims on properly identifying the reading difficulty level of a target text. To do so, the goal is to assess the various linguistic dimensions (e.g., the lexicon, the syntax, cognitive factors etc.), upon which the difficulty of a given text is based on. Due to the importance of such a difficulty estimation method, many efforts have been proposed that tried to integrate a readability assessment module on various of the most important NLP applications, such as machine translation and text simplification, rendering it in that way a decisive component of many NLP pipelines. However, the target interest of this work will mostly concern the readability assessment approaches that have been extensively incorporated in the previous mentioned domains of CALL and ICALL, for educational purposes.

A diversity of research approaches have been proposed for the task of readability assessment. Traditionally, one of the first proposed approaches of that sort, was with the use of various readability assessment formulas. Perhaps the Flesch, 1948 and Dale and Chall, 1948 are two of the most well-known examples of such a case. These are statistical models that can predict the difficulty of a text, based on simple linear regression models that combine syntactic and lexical surface text features, and targeting L1 readers. Such formulas have managed to provide some valuable results and they are still extensively applied on various domains (among which, also those of the CALL and ICALL). However, they have also been strongly criticized for their simplicity, since they only consider superficial features, ignoring other possible important aspects of readability (such as text coherence etc.).

On the other hand, many different approaches had been subsequently introduced to surpass the inconsistencies appeared on those traditional readability assessment formulas, as for example the method of Si and Callan, 2001 who presented a unigram Language Model (LM), able of identifying the readability level of science web pages by combining context-based and surface linguistic features or Collins-Thompson and Callan, 2004 who had tackled the readability assessment task in a similar setting, with a LM, using a Multinomial Naïve Bayes classifier. In addition, a diversity of ML approaches on that subject had also been introduced. Indicatively, one of the first ML approaches that were proposed was that of Schwarm and Ostendorf, 2005 where a Support Vector Machines (SVM) classifier for such a task was introduced, with the incorporation of a diverse combination of ML features, such as statistical LMs along with other traditional and surface-based ones. In that way, an open space for ML approaches on the task in hand was also introduced, where many research works have been reported since then, and are still being presented until these days.

Nevertheless, those approaches, along with the majority of the approaches on the subject at that time, had been exclusively targeting readability assessment for L1 readers, which might not be the best practice when dealing with L2 learning settings, due to many differences that may exist on the way that L1 and L2 reading

acquisition happens (as it was also indicated by Beinborn, Zesch, and Gurevych, 2012). As a result, many research examples were also reported that proposed a readability assessment perspective, specific to L2 readability. One of the first examples of such an approach might be the work of Heilman et al., 2007, who reported a much greater importance of the grammatical features leveraged in their work, for L2 readability, in comparison to L1 readability. In a similar spirit, Vajjala and Meurers, 2012 had also incorporated developmental Second Language Acquisition (SLA) measures and combined them with traditional readability features such as word and sentence length, reporting a superior importance of the lexical features over the syntactic ones, when it comes to L2 learners. In that way, besides the incorporation of different approaches to treat the task in hand, the different aspects of readability had also started to be considered as important factors that should also be taken under consideration.

In a similar sense, besides being L2 specific, readability assessment had also been treated as a task that is tightly related on the specific language being used in a target text, indicating in that way another perspective of that task. Among various L2 readability assessment approaches that have been targeting specific languages, we might report the work of Pilán, Vajjala, and Volodina, 2016 for the Swedish language, where a ML classifier was employed. In a similar sense, Forti et al., 2019, had also leveraged a ML classifier to predict the difficulty level of Italian L2 text, based on various linguistic features.

Eventually, another perspective of readability assessment that was indicated in the relevant literature and should be carefully taken under consideration, is the text structure of the written text, that is to be assessed. More specifically, most of the approaches that were discussed until now, have mainly aimed to assess readability on a paragraph if not a document level, rendering the sub-topic of sentence-level readability assessment, in a need of more research. In fact, assessing for a sentence-level readability might prove to be a much more demanding task, as it was also confirmed by Pilán, Vajjala, and Volodina, 2016, who managed to obtain an adequate accuracy at the document level for a 5-point difficulty scale, while reported a significantly lower accuracy result at the sentence-level on the same task. Furthermore, the low accuracy results at the sentence-level assessment task of Vajjala and Meurers, 2014, in comparison with the adequate accuracy scores they obtained for document-level tasks, should also be indicative points that sentence-level readability assessment might be a rather different task than a document-level one.

3.1.2 Deep learning approaches and readability assessment corpora

All of the aforementioned examples, aimed on indicating that a readability assessment task, is tightly depended on different aspects of a given text (plain language, type of exercise, etc.) and that a variety of different approaches might be possible to be incorporated (traditional formulas, ML models, etc.). In addition, as we have seen, the target learning setting (L1 or L2), the specific language in use (English, Swedish, etc.) or even the structure of the text to be assessed (document, paragraph, sentence, etc.), might only be some of the factors that could strongly affect a readability assessment procedure, rendering the task in hand a rather volatile one. Besides those mentioned aspects, we might even imagine a set of much more challenging and hardly spotted difficulty features such as the style of writing, the learner's native language and cultural background, or even the age, and so on, that might further affect a readability assessment procedure. Due to that variable nature of readability assessment, we have seen that the traditional readability assessment formulas have

been replaced with simple models that used superficial prediction features, which in their turn were surpassed by the recent and state-of-the-art ML models, since those models are capable of employing a more diverse range of complex and subtle features, rendering them capable of capturing a greater variety of necessary features.

However, such ML algorithms usually come with the burden of an appropriate *feature engineering* step to perform the best possible results, demanding for a set of fine-grained features, that should be manually defined. These features, can be rather simple (such as the sentence length or the character count) or even much more sophisticated (such as information from syntactical parse trees or semantic features), but they are in any case, mostly the result of a thorough and experienced research from a group of domain-experts. Therefore, since they comprise a vital step for any ML algorithm, a ML model appears to be greatly depended on them and the results will only be as good as the acquired features. Therefore, concerning the task in hand, it might be easily understood that the variety and the subtle nature of all the different possible features that were previously mentioned, might render such a feature engineering process even more difficult. Thus, to overcome all the issues that might derive from a procedure, DL models have made a massive appearance during the last recent years, and they have managed to perform very well on many of the previous ML tasks. The reason for that, is that these models can perform very well without any need of a feature engineering process, eliminating the necessary domain of expertise and instead, they only require for a large amount of quality training data and a rather efficient model complexity, to resolve any task.

As a result, DL models have also been employed in the research field of readability assessment, even though the research findings for the moment might considered to be rather scarce. For example, Lo Bosco, Pilato, and Schicchi, 2019 presented a Recurrent Neural Network (RNN) model that automatically detects whether an Italian sentence is simple or complex. On the other hand, Martinc, Pollak, and Robnik-Šikonja, 2021 had leveraged much more complex architectures and managed to achieve state-of-the-art performance with a pre-trained Transformer model and a Hierarchical Attention Network (HAN) model, two complex and efficient DL architectures, in two well-known readability assessment corpora. Furthermore, Deutsch, Jasbi, and Shieber, 2020 had also reported promising results on that subject. In that work, it was indicated that there exists no improvement at all with the incorporation of linguistic features in DL methods, signifying in that way, that these DL models might already be able to implicitly capture the features that are useful for readability assessment.

However, most of the research findings on DL readability assessment, in our knowledge, are very obscure when it comes to readability assessment for L2 educational contexts. Most of the existing L2 readability assessment approaches restrict themselves to the traditional or the ML methodologies that we have previously reported, leaving an open research space on ways that DL models could also be incorporated for that task. Nevertheless, for that to happen, it is necessary that a set of freely available and extensive L2 readability assessment corpora might exist, since these DL models are intensively data-depended, as we have mentioned, and they need such resources to be trained on.

Some examples of such L2 readability assessment corpora that have been proposed in the relevant literature, might be the cases of Schwarm and Ostendorf, 2005 and Vajjala and Meurers, 2012, with the *WeeklyReader* and the *WeeBit* corpus, accordingly, or the *Newsela* corpus that was proposed by Xia, Kochmar, and Briscoe, 2019. However, even though those corpora might be rather useful options for many L2 readability assessment tasks, they come with the burden of being annotated on

a document or a paragraph level. As a result, for settings were sentence-level annotated corpora are necessary, they might prove to be rather unsuitable, since the structure of the target text is also an important aspect of readability, as it was previously illustrated. Even though one might think that such a corpus shortage could be surpassed with the employment of models that are trained on a document-level, to sentence-level tasks, it has been thoroughly indicated by Skory and Eskenazi, 2010 that document-level assessment methods would be unreliable for shorter texts, such as single sentences, indicating in that way the necessity for readability assessment corpora that are exclusively annotated on a sentence level.

A scarce number of research approaches for that matter have been reported. Many of these approaches had tried to generate a corpus with the leverage of the Wikipedia and the Simple Wikipedia databases, where sentence-alignment procedures might be leveraged between the sentences of these two databases. Such sentence-aligned corpora have mostly been used for binary readability assessment tasks on the sentence-level, where the Simple Wikipedia sentences are considered as easier targets in comparison to the Wikipedia ones, and custom labels that correspond to that distinction are created. Among many existing approaches on the creation of such aligned corpora, we might mention the approaches of Zhu, Bernhard, and Gurevych, 2010 and Hwang et al., 2015, indicatively. Nevertheless, this way of approaching L2 sentence readability tasks, is restricted on incorporating only a binary range of difficulty levels, rendering that method unsuitable, for tasks where more extensive difficulty scales might be needed.

Eventually, some scarce sentence-level L2 readability assessment corpora have also been proposed. Indicatively, we might reference a very famous and recent one, namely the *OneStopEnglish* corpus (Vajjala and Lučić, 2018). That corpus was initially compiled at the document level, retrieving its material from an online English language learning resource. Each retrieved article was rewritten by English teachers to suit three language levels for adult L2 learners, with those being Elementary, Intermediate, and Advance, resulting in that way a manually annotated L2 document-level readability assessment corpus. Subsequently, that corpus was converted from the document-level into a sentence-level, with the aid of the cosine similarity measure. In the end, the final corpus that was acquired, was a sentence-aligned corpus of 6,994 sentence pairs. Nevertheless, such a size of a corpus might be quite limited, for the needs of a modern DL model. In a similar sense, some other scarce examples of such sentence-annotated corpora might exist, which are also, in our knowledge, quite limited on their extent, imposing in that way a great restriction when it comes to the data-driven DL approaches that were previously mentioned.

3.1.3 Conclusion on readability assessment

Given all that, we too have decided to employ the advantages of the DL techniques in our work. In that way, we may be able to overcome on the biggest possible extent, most of the issues that might emerge from a typical feature engineering procedure, while also identify some of the most subtle difficulty features. More specifically, we have treated the readability assessment task as a multi-class classification problem where the task of our DL model, is to learn how to classify a given sentence in an appropriate difficulty label. In our intuition, such a DL model may be capable of automatically identifying all the necessary features that should be manually defined in the case of a ML approach, and perform a classification procedure that might be intuitive enough even for a human.

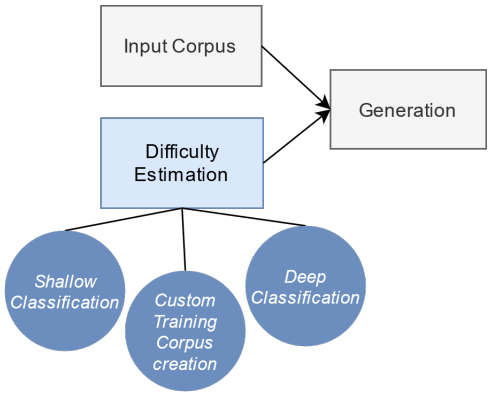


FIGURE 3.1: The components of the Difficulty Estimation module.

However, these DL techniques, as we have mentioned, are tightly depended on the available training data, based on which any necessary dimension for a target task, could be inferred. Given that our findings on such L2 sentence-level readability assessment corpora were rather limited, we have further created our own custom training corpus, so that the DL model that was leveraged, could be trained on. A two-fold classification approach was developed for that to happen. An initial *Shallow Classification* was used to create a custom training corpus and then, the DL model was trained on that custom dataset with a second, *Deep Classification* step. For the custom training corpus to be created, we have leveraged an extensive set of English sentences, along with a difficulty graded lexicon. Having those two resources, each one of those sentences were automatically annotated based on the entries of the graded lexicon, creating in that way the custom training corpus.

As a result, the Difficulty Estimation module is consisted of three major components, namely an initial *Shallow Classification* component, a *Custom Training Corpus creation* component, and a *Deep Classification* component. The overall conception of this module is being demonstrated on Figure 3.1, and a thorough presentation of all its details will be presented on the next part of this Chapter.

3.2 Readability assessment on SemiGramEx

3.2.1 Custom training corpus creation with a Shallow Classification

Having reported an overview of the existing relevant literature for the subject of readability assessment, as well as the general conception of our approach, we should now more thoroughly elaborate on the methodological decisions and details that were followed for that task.

The first deficiency that occurred in this part, was the absence of sufficient sentence-level annotated corpora to train our DL model, since even the scarce previously mentioned examples, were far from being adequate for a modern DL model due to their limited extent, as it was already mentioned. To overcome that issue, we decided to create our own custom training corpus. However, manually annotating a corpus is a rather time consuming and expert-knowledge-depended process and therefore, we aimed on creating an annotated training corpus in an automatic way. To compose such a training corpus two things were required, those being an extensive and suitable set of *English sentences* and a set of *corresponding difficulty levels* for each one of those sentences.

An important source of inspiration at that point had stand the work of Lee and Luo, 2016. In this work an automatic generation system for preposition exercises was introduced, where the difficulty of each sentence was estimated with respect to its vocabulary, using graded vocabulary lists and a frequency corpus. Even though that approach might be conceived as rather simplistic, at least in respect to the ML ones which we were trying to avoid, the intuition that the vocabulary of a sentence (based on a predefined manually-graded lexicon, rather than any specific, linguistically oriented lexical aspect), could demonstrate some sort of difficulty classification validity, had been the starting point for our thinking on that subject. More precisely, we have initially followed a similar way to create the custom annotated training corpus in such a vocabulary-based manner. That lead us to the conception of the first component of this module, namely the *Shallow Classification* component, where a first-step classification of English sentences solely based on their vocabulary was performed, to create the training corpus.

To implement the Shallow Classification component, we had initially leveraged the *CEFRLex graded lexicon* (Dürlich and François, 2018) as the main source of our custom annotation process. The CEFRLex lexicon is a graded resource that was created through international collaborations between a variety of research groups, specialized in linguistics and language acquisition and for the moment it hosts 5 languages, with English being one of them. The English corpus contains 15,280 words, that are described in terms of word usage in pedagogical contexts, over the CEFR scale. For each word, a frequency distribution in the range of A1-C1 CEFR levels, is presented. A snapshot of this lexicon is demonstrated on Appendix A. Furthermore, it should be mentioned at this point that, While several other possible resources of that sort might also had served for the task in hand (like any well-known word frequency list, such as the Google n-gram corpus[1]), such resources are usually made for generic-purpose tasks and they mainly target native-speaker language. On the contrary, the CEFRLex lexicon is a resource that targets L2 learning, teaching, and research, a fact of a great importance for our task.

To proceed from the word frequency distributions that are presented on the CEFRLex lexicon, to the assignment of a specific difficulty label on each word entry, we followed an approach similar with Soler, Apidianaki, and Allauzen, 2018. More specifically, we assigned on each word entry of the lexicon the CEFR level of its first appearance, meaning that a word would be considered on a given CEFR level, if that CEFR level would be the first time that a frequency score might appear on the graded lexicon for that word (indicating also in that way the moment of acquisition). Moreover, to make the difficulty categories as distinct and robust as possible, we clustered the 5-scale CEFR labels, presented on the graded lexicon, into three difficulty categories. To do so, the A1 and A2 CEFR levels were packed together into a *Beginner category*, the B1 and B2 were packed into an *Intermediate category* and the C1 was held as an *Advance category*. In that way, we managed to eventually acquire a graded lexicon, containing around 15k words that were labeled in three difficulty labels.

Consecutively, as we have mentioned, the other necessary component to create such a custom corpus was a set of proper English sentences which would be annotated based on the CEFRLex lexicon. The main obstacles that were confronted when searching for such a collection, were the necessity for it to be publicly available and to be structured in the form of sentences. That means we avoided incorporating existing English text resources structured in a paragraph or a document level and

[1] https://books.google.com/ngrams/info

convert them into a collection of sentences, since possible license issues might occur with that approach (a license provided for the whole data, might not be applied at each sentence of it, separately). Furthermore, we also wanted to avoid the conversion of a document into sentences, to eliminate any possible discrepancies that might occur by a sentence split parser (since in that way, many of the sentences being split, might be intensively context-dependent and result error-prone training data). In addition, the language usage on any such corpus should be generic and of every-day use, to avoid any bias on specific topics or language forms.

Furthermore, a last point that was important for the selection of English sentences as a raw training material, was the amount of them. As we have already mentioned, the CEFRLex corpus only contains around 15K word entries and therefore it cannot be considered as an extensive annotation resource. As a result, it would most likely be the case that a decent number of our collected sentences would have had to be skipped, due to many contained words without an entry in our lexicon (since in that case it would not be possible to annotate such sentences). As a result, we wanted to incorporate a collection of sentences that would be lengthy enough, so that a surplus of sentences could be discarded if needed, while also maintaining a threshold of necessary sentences for our collection.

To the best of our knowledge, the publicly available English sentence-level corpora that fulfill all the above criteria are not abundant and therefore, small compromises had to be done on the quality of it. As a result, we had selected as the best candidate for that task the collection of English sentences provided by the *Tatoeba database* [2], since we observed a rather optimal behavior of it with our model, in comparison with other candidate corpora resources that we tried.

The Tatoeba project, is self-introduced as a collection of sentences with their translations, that is collaborative, open, and free. The project was started at 2006 and is released under the CC-BY 2.0 FR.5 license. It is a multilingual project that hosts for the moment around 405 different languages, aiming also on languages that are low resourced. The Tatoeba database is becoming richer every day with the means of crowdsourcing, based on which anybody can propose new sentences and translations in a target language or translate existing sentences. It contains a range of simple to complicated sentences and its aim is to provide example sentences along with their translations, for various linguistic constructions in different languages. In addition, Tatoeba was initiated and is still oriented mainly on L2 learners, a fact that was very important for our purpose.

For the task in hand, we leveraged the Tatoeba collection of English sentences without their translations. In that way, we had in our disposal a set of around 1M English sentences that were well-distributed across different sentence lengths. In addition, to overcome in the greatest extent the bias and the low quality that usually follows such collaborative tasks, we leveraged a list of subscribed translators that is provided by Tatoeba. In that list, the proficiency level of each translator was reported and therefore, we filtered the collected English sentences to only maintain sentences that were proposed by translators with the highest rating level. In that way, we only retrieved the best possible results in terms of language fluency and quality.

Having acquired such a collection of English sentences from the Tatoeba database, as well as the CEFRLex difficulty graded lexicon, the next step of our implementation was to employ those two resources to create our *Custom Training Corpus*. As it was already mentioned, the creation of the Custom Training Corpus was conducted with an initial, Shallow Classification step. More specifically, for each word in each

[2] https://tatoeba.org/en/

of the collected English sentences, we would assign their difficulty labels, according to the entries of the CEFRLex lexicon. Then, the highest difficulty label that was assigned to the words of that sentence, would be assigned as the difficulty level of the sentence itself. Only Verbs, Nouns, Adjectives and Adverbs were considered in each sentence as possible candidate tokens for that part, mainly because those were the main part-of-speech tags existing in our graded lexicon, but also since we considered them as the most important candidates. That means, for a sentence to be assigned with a label, all its words, that correspond to the four mentioned part-of-speech tags, had to simultaneously exist in our graded lexicon. In that way, we avoided the possibility of naively classifying sentences that contained words, unknown to our lexicon.

3.2.2 Deep Classification

Having created our custom training corpus in such a shallow way, we then trained a DL model as a second, *Deep Classification* step. While we could only follow an approach similar to Si and Callan, 2001, that was previously mentioned, and simply stick on classifying the difficulty solely based on the vocabulary of each sentence, we were restricted on doing so, since the graded lexicon we had incorporated was quite limited to classify any possible unknown sentence (i.e., a sentence without an entry on the graded lexicon, would not be possible to be assigned with a difficulty label). Furthermore, a DL model should be in any case a better solution than a classification that is only based on the vocabulary, since such models are able of capturing a variety of subtle features for a given task. Therefore, our intuition was that if we would provide on a DL model, a Shallow Classification as an initial guidance (in our case the difficulty labels of the custom annotated corpus, which were based on the graded lexicon), then the model might be able to learn on its own all the rest, underlying difficulty factors, that might be required to properly generalizing on any unknown sentence. The empirical results of our efforts during the testing stage, indicated that our intuition was valid enough and therefore, we decided to follow that solution. The overall two-fold classification process, is demonstrated on Figure 3.2.

Concerning the DL model architecture that we employed for that purpose, that was a bidirectional Long Short-Term Memory (biLSTM) one, since we found that this architecture was sufficient to capture any non-linear features' dependencies and underlying language difficulty factors, that we were hoping for. As we have already mentioned before, we treated the task in hand as a 3-label multiclass classification problem and we initially experimented with different layer sizes and a diversity of other architecture features or hyperparameters, before we conclude to the most adequate candidates. All the selected hyperparameters, were chosen either with an empirical evaluation of their results, or as the most frequently used choices in the relevant literature for such DL architectures.

More specifically, the hidden size of the model was chosen to be 64 and the pretrained 300-dimensional Glove word embeddings (Pennington, Socher, and Manning, 2014) were incorporated, to encode any necessary distributional information of each word. In addition, a dropout of 0.8 was employed to help the model generalize more accurately and the ReLU activation function was used. Subsequently, we chose CrossEntropyLoss as the most relevant loss function for our task and Adam as an optimizer.

In addition, during the creation of our custom training corpus, we balanced our 3 classes to 69,900 sentence instances on each one, since above that limit we noticed

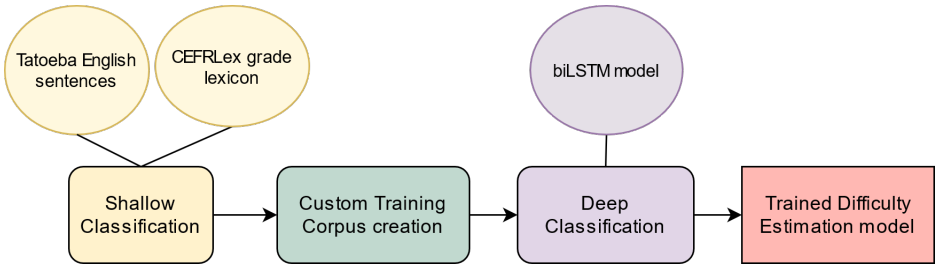


FIGURE 3.2: The two-fold classification process to create a custom training corpus and to train a DL classifier.

that the biLSTM model was overfitting. The training corpus was properly prepossessed and split into a training and a test part, with 25% of the overall corpus being incorporated as a test part. We trained the model for 7 epochs, with a batch size of 128 and a learning rate of 0.001, applying also a 5-fold splitting. The training rates, along with a further evaluation of our model's results will be subsequently presented on Chapter 5, where the overall evaluation of our work will be reported.

Chapter 4

The Generation module

Both of the modules that have been reported until now, were created on an initial step of the implementation procedure and they were mostly presented as independent modules so far. In this chapter, the final module of our framework will be presented, namely the *Generation* module, which is the component that is responsible of carrying the main generation procedure of SemiGramEx, with the simultaneous incorporation of the previous two modules' results. More precisely, the Generation module foresees on automatically generating a set of L2 English grammar exercises based on some input generation parameters, such as a target teaching goal and a difficulty level, among others, that correspond to aspects of the previous two modules' results.

Following in this Chapter, we will initially present some of the existing L2 exercise generation approaches in the literature, that are relevant with the task in hand. The goal of such a retrospection, is to indicate the main generation approaches that have been adopted for that task, as well as other aspects of it, such as the different types of exercises or the different generation resources that have been leveraged. Once we acquire a clear view on the overall literature of the subject, we will subsequently present all the details of the implementation that we have followed to create the Generation module. Eventually, the overall SemiGramEx framework interface will also be presented, so that the reader might have a clear picture of it.

4.1 Relevant literature

One of the first research proposals that was focused on grammar exercises generation was the work of Chen, Liou, and Chang, 2006, who introduced the *FAST* generation system. In that framework, a semi-automatic generation method for grammar exercises was developed and text material retrieved from the Web were leveraged as a generation basis. A set of handcrafted patterns and rules were leveraged to transform that text material into appropriate grammar exercises, based on which, sentences retrieved from the Web were converted into Multiple-Choice[1] and Error Detection types[2] of questions.

Subsequently, a more recent approach on that subject might be the approach proposed by Soonklang and Muangon, 2017. In that work, a web-based system was implemented to generate English exercises for L2 students in an automatic manner. L2 English textbooks were leveraged and processed with NLP methods, so that they would be split into sentences, and those sentences were in their turn automatically annotated with relevant linguistic information. Having acquired such annotated

[1] A type of exercise where the retrieved sentences, were converted into demonstrating erroneous statements mixed with correct ones, as possible solutions, for a target grammatical construction.

[2] A type of exercise where the retrieved sentences, were converted into demonstrating wrong statements in the place of the original correct ones, for a target grammatical construction.

sentences, relevant subsets of them were targeted, based on specific grammatical goals, and those sentence subsets were subsequently converted to a given type of an exercise with a rule-based approach. More specifically, four different types of exercises were supported, those of the typical Fill-in-the-blank[3] exercise, the Either/Or[4] exercise, the True or False[5] exercises and the Error Correction[6] type of exercise.

In addition, besides English as a target language, various research proposals for the task in hand have also been proposed for other languages. One example of such a case might be that of Chalvin, Eensoo, and Stuck, 2013 for Estonian, where French-speaking learners were targeted. In that example, an Estonian-French parallel corpus had been leveraged as a generation basis, along with a custom difficulty classification approach and a set of content selection criteria. More specifically, the system was able of retrieving relevant sentences from the existing parallel corpus and converting them into appropriate Fill-in-the-blank grammar exercises, with handcrafted transformation rules.

However, even though the approaches that were presented so far had extensively incorporated a dedicated input resource as a generation basis, there have also been some scarce research proposals that followed a different path. A characteristic case of such an example might be the work proposed by Perez-Beltrachini, Gardent, and Kruszewski, 2012, namely *GramEx*. In that approach, instead of employing a pre-defined corpus as a generation basis, the authors had created a system that was able of directly generating grammar exercises in a controllable manner, concerning both aspects of lexical and syntactical complexity. To do that, a formal grammar was leveraged, where the syntax and the semantics of the allowed exercise generation patterns were described, and in that way, the system was able of generating a set of exercises targeting a specific pedagogical goal, under specific conditions. In more details, typical surface realization techniques were initially used to construct a set of candidate sentences. Then, the most relevant of those sentences were retrieved, based on a pre-defined constraint language. Having acquired such sentences, those were then converted into appropriate grammar exercises, with the use of rule-based techniques. Two types of exercises were supported by GramEx. The first one was a typical Fill-in-the-blank exercise type and the second one was a Shuffle question exercise[7].

Eventually, a research work that should be presented before we conclude this overview is the one that was recently proposed by Lee and Luo, 2016. That work is also the most similar one with our approach, even though the target users in that case were L2 learners, rather than L2 teachers. More specifically, Lee and Luo, 2016 had introduced a system that can automatically construct a set of Fill-in-the-blank grammar exercises. The system was specifically targeting *preposition learning* and the Wikipedia database was incorporated as an input resource for the generation procedure. Initially, the Wikipedia sentences were linguistically annotated with a state-of-the-art parser and then, various rule-based matching queries were performed, to retrieve candidate sentences that were adequate for the task in hand. Those sentences were eventually converted on the appropriate exercise form, in a rule-based manner.

[3] A type of exercise where the target grammatical form is replaced with a blank and its corresponding lemma is presented as a hint to the learner.

[4] A type of exercise where a correct answer between two choices must be chosen.

[5] A type of exercise where the learner must also correct the false question, besides identifying it.

[6] A type of exercises where the incorrect word in a sentence must be spotted.

[7] A type of exercise where the function words of a sentence are deleted, and the remaining lemmas are shuffled, before being demonstrated to the learner.

To conclude, as it might have already been observed, the generation approaches that were presented so far for the task of automatic generation of L2 grammar exercises, have primarily been rule-based ones. More precisely, the most usually adopted approaches had leveraged an input resource, which was processed in a relevant way, so that appropriate text material might be retrieved. Such text material in their turn, could be converted into appropriate grammar exercises. In fact, besides the aforementioned examples, in our knowledge, the majority of the research approaches for that task had followed similar paths. Therefore, we too have decided to leverage a rule-based approach, similar to those that were previously mentioned.

Nevertheless, it should be noted at this part, that this decision had in fact greatly troubled us during the implementation of this module. More precisely, most of the research trends in all of the NLP domains nowadays, are extensively leveraging DL models to solve any task. In a similar sense, for a task like ours, some of the typical *Automatic Text Generation* techniques could be incorporated (such as the employment of a sequence-to-sequence architecture etc.). Therefore, we had greatly troubled ourselves on whether we might also follow such a text generation approach. In the end, mostly due to the research literature that was previously reported, we estimated that there was no need of employing complex DL architectures for the task in hand. However, we might point out that possible future research on that subject might prove to be rather useful.

4.2 Generating grammar exercises with SemiGramEx

Having introduced some of the most frequently reported approaches on automatic L2 grammar exercises' generation, we have now arrived at the final point of our implementation pipeline, that being the *Generation* module.

To recap what has been done until now, we have already seen that SemiGramEx initially incorporates an input resource as a generation basis. That resource is pre-annotated with a plethora of linguistic information, retrieved by a state-of-the-art parser, so that specific subsets concerning a target teaching goal might be easily isolated. In addition, we have trained a DL model that is capable of classifying English sentences in a range of three language proficiency levels. That DL model was then leveraged to label with the corresponding difficulty labels, all the existing sentences on the input resource in use. In that way, the input resource is annotated both with the relevant linguistic information and with the corresponding difficulty labels, and the Generation module of SemiGramEx can perform a set of queries on that input resource to retrieve sets of relevant sentences that correspond both to a target teaching goal and a target difficulty level. Having acquired such sentences, those can then be converted into appropriate grammar exercises with the use of rule-based techniques and hand-crafted transformation patterns. An illustration of the above can be seen in Figure 4.1.

Following in this section, we will firstly present the types of exercises that exist in our generation framework, along with the target teaching goals that are supported. Eventually, the overall SemiGramEx interface will be presented and a small recap on all the existing components will be made, so that the system will be introduced as a complete framework. In the Appendix B, a pivot table of all the details that will be subsequently reported is also presented, to facilitate the overview of all the existing generation parameters.

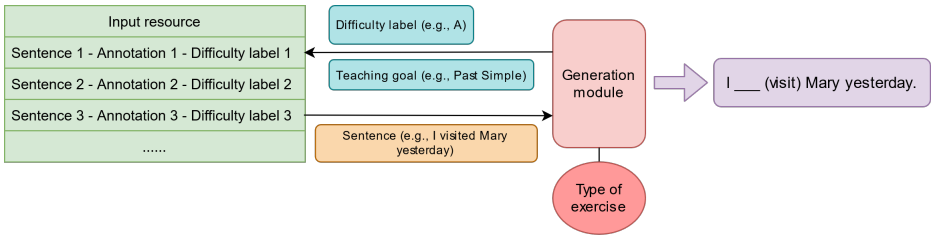


FIGURE 4.1: A more detailed exercise generation process on the Generation module.

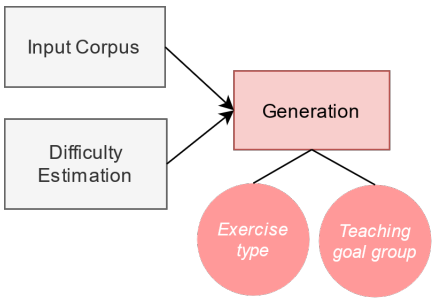


FIGURE 4.2: The Generation module of SemiGramEx.

4.2.1 Types of exercises and teaching goals

SemiGramEx supports for the moment three different *types of grammar exercises*, those being the *Fill-in-the-blank*(FIB) type, the *Multiple-choice* type and the *Find-the-mistake* type of exercise. In addition, it exclusively targets *verb tenses learning*, where the supporting verb tenses are those of *Present Simple*, *Present Progressive*, *Past Simple*, *Past Progressive*, *Present Perfect* and *Past Perfect*. Those six verb tenses are further organized in a set of three *teaching goal groups*, that will be subsequently presented, where each one of those groups represents a supported teaching goal. Nevertheless, all these types of exercises and teaching goal, are only supposed to be an initial part of the generation toolkit that should be further extended in the future. In Figure 4.2, the main components of the generation module are demonstrated.

Starting from the supported verb tenses, those have been mostly selected in accordance with the linguistic constructions that existed in our input resources. More precisely, it was observed that the two incorporated input resources (the Simple Wikipedia one and the BNC one), were inadequate on supporting every existing type of verb tense. For example, verb tenses such as the *Future Perfect* or the *Present Perfect Continuous*, among others, were not actively present in any of the two generation resources and therefore, their employment was not possible. The reason for that, has to do with the type of language and the genres or topics that are mostly leveraged in our input resources, meaning that some verb tenses would appear more frequently on some language settings than others. As a result, we were restricted on employing only the six aforementioned verb tenses for the generation of our grammar exercises.

Subsequently, concerning the supported teaching goal groups, those are the *Present Simple and Present Progressive* group, the *Past Simple and Past Progressive* group, and the *Present Perfect and Past Perfect* group. The reason we have chosen to demonstrate the supported verb tenses organized in those three groups, was to provide a better flexibility on the final generated results. More precisely, we wanted to support two exercise generation options, with the first one being the generation of ready-made

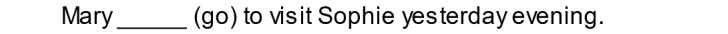


FIGURE 4.3: Example of a FIB type of exercise for the past simple tense.

exercises that target the simultaneous teaching of each one of the verb tenses that exist in a teaching goal group, and the latter being the generation of exercises that specifically target in isolation the teaching of each one of the verb tenses existing in a teaching goal group. To do that, we have employed an extra shuffling option, that will be more thoroughly presented subsequently, where the user can choose whether the generated exercise instances should be demonstrated in a random order or not. Having chosen a target teaching goal group, a type of exercise, and a specific number of exercise instances to be generated, SemiGramEx generates half of the exercise instances number for the first verb tense that exist in the target teaching goal group and the other half for the second verb tense of that group. Eventually, those exercise instances are presented as separate sets of exercises, targeting separate verb tenses, or as mixed ones, depending on the shuffling option we have mentioned.

To understand more thoroughly the above, we might consider a generation scenario with *FIB* being the exercise type, *Present Simple and Present Progressive* being the teaching goal group and *10 exercise instances* having been chosen to be generated. In such a scenario, SemiGramEx would generate five FIB exercise instances for the present simple tense and five FIB exercise instances for the present progressive tense. Those exercise instances would then be demonstrated as two separate sets for each one of the target verb tenses, targeting the teaching of each one of those verb tenses separately, or they could be demonstrated as a mixed set of exercise instances, targeting the teaching of both tenses in a mutual setting. The way those exercises would be demonstrated, depends on the shuffling option.

Eventually, concerning the supported exercise types, the first one that was incorporated in our framework was the previously mentioned FIB one. The FIB type of exercise is perhaps one of the most frequently used exercise types that a L2 learner might have to confront in a L2 grammar learning setting. It is built with the removal of a target word from a sentence (which concerns a target teaching goal) and the replacement of it with a blank, followed also by an indication of the blanked word's lemma form. The aim of such an exercise type in our system is to practice and evaluate *verb tense construction*, where the learner is evaluated on the construction of a target verb in the appropriate form. An example of such a type of exercise can be seen on Figure 4.3, for the past simple tense.

To generate the FIB exercises, SemiGramEx performs several queries iteratively on random subsets of the target input resource, so that only sentences containing a verb in the target verb form are retrieved. That iteration continues, until a set of sentences corresponding to the target number of exercise instances to be generated, are acquired. That subset must contain sentences with at least one verb existing in any of the verb forms included on the target teaching goal group. An equal number of sentences are retrieved for each one of the target verb tenses in the teaching goal group. Having acquired a set of such sentences, SemiGramEx then identifies the index position of the target verbs in each one of those sentences (as well as the depended auxiliaries if any) and converts those sentences into FIB exercises. More specifically, the target verbs along with any possible auxiliary are deleted and being replaced by a blank, while also the lemma forms of those replaced target verbs are demonstrated in parenthesis to the learner, as a hint. All the aforementioned

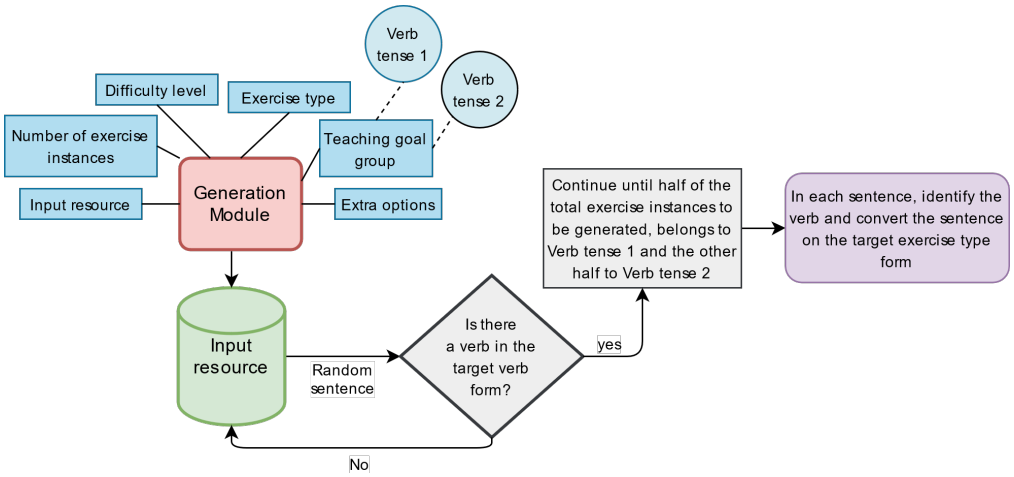


FIGURE 4.4: The detailed process of retrieving sentences before converting them into exercises.

Mary _____ to visit Sophie yesterday evening.
A:went
B:was going

FIGURE 4.5: Example of a Multiple-choice type of exercise for the past simple tense.

processes are performed with the help of dedicated hand-crafted rules and the sentences are converted with similar transformation patterns. An illustration of that procedure is demonstrated on Figure 4.4

The second exercise type is the *Multiple-choice* one, which is also a frequently used exercise type in L2 grammar learning. It is a similar exercise type with the FIB one, but in this case, instead of a lemma form, a set of possible solutions are demonstrated as a hint to the learner. More specifically, given a sentence, a target word in that sentence (that is relevant to a teaching goal) is identified and being replaced with a blank. Then, a few options are demonstrated to the learner as possible solutions. One of those options is the initially blanked word (the correct option) and the other options are wrong alternatives that are demonstrated to distract the learner (called the distractors). In our system, the target words of interest that are being replaced with a blank are the verbs, and the aim of such an exercise type is to practice and evaluate *verb tense identification*, where the learner should properly identify which verb tense to use in a target sentence. An example of such an exercise type is demonstrated on Figure 4.5, for the simple past tense.

To generate such an exercise type, SemiGramEx performs a similar procedure with the one previously presented for the FIB exercises. More specifically, a set of relevant sentences are retrieved from a target input resource, in a similar manner with the FIB exercises. Subsequently, to convert those sentences into Multiple-choice exercises, the target verbs along with any possible depended auxiliaries are identified in a similar manner with the FIB exercises, and they are replaced with a blank. Eventually, the blanked target verb is demonstrated as a possible solution along with a distractor. The original inflection of the target verb is maintained on the correct solution option and only one distractor is provided for the moment. More precicely, the distractor that is demonstrated by SemiGramEx is the target verb, inflected on the other verb tense that exist on the target teaching goal group.

To clarify that, we might consider as an example the generation scenario of a

Mary was going to visit Sophie yesterday evening.

FIGURE 4.6: Example of a Find-the-mistake type of exercise for the past simple tense.

Multiple-choice exercise type and *Present Simple and Present Continuous* as a target teaching goal group. In that case, SemiGramEx will initially retrieve a set of sentences containing a verb in the present simple verb form, where that verb will subsequently be replaced with a blank, before being demonstrated as an option to the learner. Subsequently, the same verb will be converted on the present continuous form, and it will be demonstrated as a distractor option (maintaining the correct person and number). In a similar manner, the reverse will happen for a sentence with a target verb on the present continuous tense.

Eventually, the final supported exercise type is that of the *Find-the-mistake* exercises. That exercise type is constructed by converting well-formed sentences into ill-formed ones, with the injection of appropriate mistakes on them. Having such ill-formed sentences, those can be subsequently demonstrated to the learner, in a mixed collection of ill-formed and well-formed examples. Many possible variations of such an exercise type might exist, where in one case the learner might just have to indicate whether a sentence is correct or not, or in another case, the learner might have to specifically specify the existing mistake and correct it, etc. An example of such an exercise type can be seen on Figure 4.6, where the target teaching goal is past simple learning.

Concerning the Find-the-mistake exercise type that is incorporated in this work, it is mainly oriented on grammatical mistakes and more specifically, on grammatical mistakes about verb tenses. The generation procedure that is followed by SemiGramEx is relatively similar with the aforementioned ones for the FIB and the Multiple-choice exercise types. More specifically, a set of sentences are retrieved from a target input resource, similarly with the FIB exercises, and a target verb along with its corresponding distractor are identified, in a similar manner with the Multiple-choice exercises. However, the difference in this exercise type comparing to Multiple-choice ones, is that the target verb in the correct verb tense is replaced by the target verb in the wrong verb tense inside the sentence itself (instead of being demonstrated as a possible solution along with the correct one).

To elaborate more on that, we might consider as an example, the generation scenario where a *Find-the-mistake* exercise type is selected, along with a number of *10 exercise instances* and *Present Simple and Present Progressive* as the target teaching goal group. In such a case, half of the sentences, that were retrieved by the input resource to be leveraged as exercise instances, will contain a target verb in the present simple verb form and in those sentences, that target verb will be substituted with the same verb in the present progressive form. For the other half of sentences, the reverse will happen. Those ill-formed sentences will eventually be presented as final exercise instances.

Conclusively, we should also mention that the Present Perfect and Past Perfect teaching goal group is not demonstrated as an option for any of the Multiple-choice and Find-the-mistake type of exercises, since a qualitative analysis on the generated results had indicated that two verb tenses existing in that teaching goal group, may not be easily distincted in such exercise types and therefore, it might be optimal not to support them (more details about that on Chapter 5, where the evaluation part of our work will be presented).

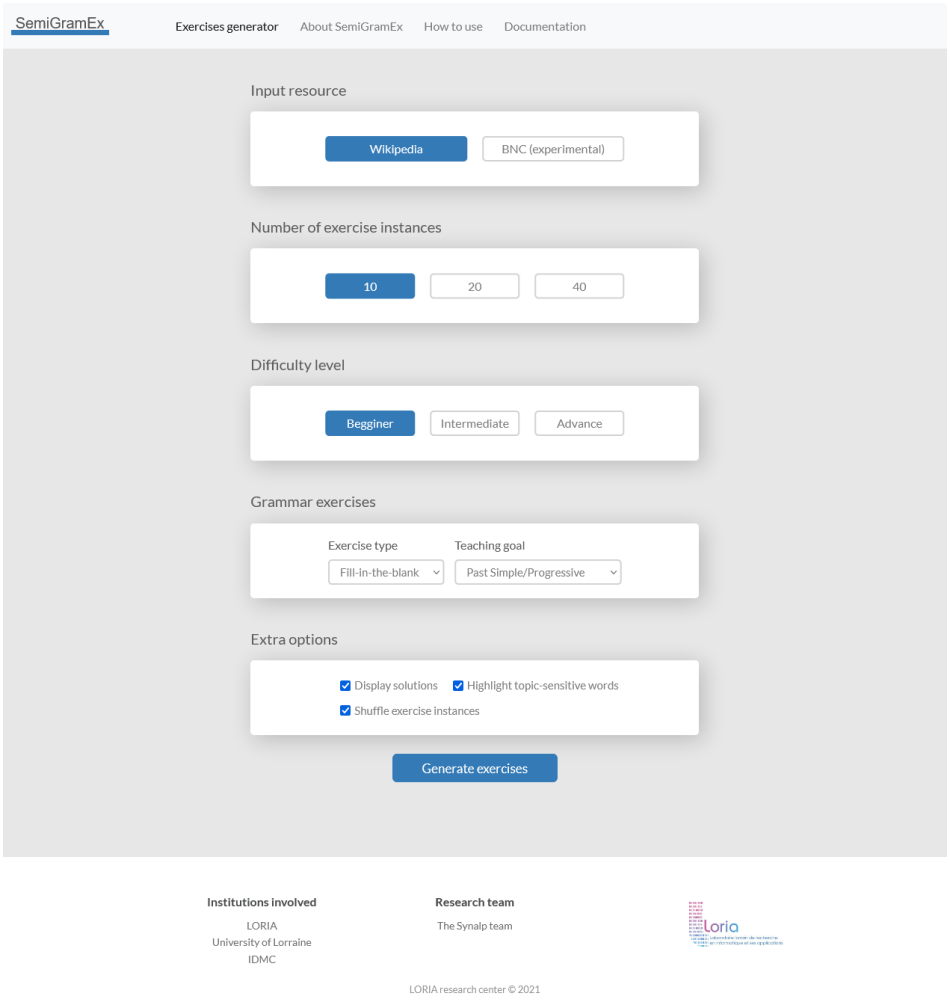


FIGURE 4.7: The SemiGramEx web-interface.

4.2.2 Overall presentation of the SemiGramEx user-interface

At this point, we have already introduced all the different components that compose our generation framework. Therefore, we will now conclude the implementation part of our report, with an overall presentation of the SemiGramEx user-interface that bonds together all the aforementioned components, while also presenting a brief recap on each one of them. The reason for that, is to present the complete picture of our framework before proceeding on the evaluation part of its results. However, it should be kept in mind that the existing interface might only be a temporary one, since the long-term goal of this work is to develop a fully autonomous L2 generation toolkit, where novel and specific interface guidelines might have to be followed then. The SemiGramEx interface is demonstrated on Figure 4.7. In addition, a table of all the main supporting parameters is demonstrated on Appendix B for an extra perspective.

Starting with the introduction of the web-interface, all the necessary information that surround the SemiGramEx project, are contained on the Header and the Footer parts of the web-interface. More specifically, on the Header part, the user might have the chance to find information about the general context of this project, tips about the meaning of all the interface's components and how to use them, as well as an extensive documentation where the whole implementation procedure is thoroughly

explained. On the other hand, on the Footer part of the web-interface, information concerning the institutions that were involved for this project to happen, as well as references about the host research team, are also presented.

Subsequently, concerning specifically each one of the generation components that exist on the web-interface:

- **Input resource**: That parameter concerns the input resources that are incorporated as a generation basis of SemiGramEx and were reported on Chapter 2. The Wikipedia option utilizes the Simple Wikipedia as a generation basis, while the BNC option utilizes the small British National Corpus resource as a generation basis. However, it should be noted that the results of the BNC corpus are still not optimal, and this option is only provided for a plurality of the generation results, under the assumption that it is still on an experimental stage.

- **Number of exercise instances**: That parameter concerns the number of final generated exercise instances. Only three possible options are allowed for the time being (those of 10, 20 and 40 exercise instances), to eliminate the processing time that is required by the system during the generation process. In a future scenario where sufficient processing power could be obtained, an undefined number of exercise instances might be leveraged.

- **Difficulty level**: That parameter concerns the language proficiency level that the final generated exercises, should maintain. The process of creating such a difficulty estimation component, was thoroughly reported on Chapter 3. Three difficulty levels are supported for the time being, with those being the Beginner, the Intermediate and the Advance proficiency levels and the aim of that generation parameter is to aid the teacher-user, by classifying the generated results in one of those three difficulty levels. However, it should be mentioned that during the implementation of this module, it was strongly assumed that a L2 teacher must interfere and improve any possible deficiency of the difficulty estimation results, meaning that the results of this classification component, might not be necessarily sufficient to be directly demonstrated on a L2 learner. In addition, the difficulty classification module that is incorporated in this work, is not meant to concern L2 learners with an absolute beginner level of language, but rather learners with some initial, even though poor, language background.

- **Grammar exercises**: That parameter concerns the nature of the final generated grammar exercises. More specifically, the *Exercise type* option, concerns the type of the generated exercises, with those being for the moment the Fill-in-the-blank, the Multiple-choice and the Find-the-mistake types. Furthermore, the *Teaching goal* option, concerns the target teaching goal group, with those being the Present Simple and Present Progressive group, the Past Simple and Past Progressive group, and the Present Perfect and Past Perfect group. More details about that parameter, were previously reported on this Chapter.

- **Extra options**: These parameters aim on further helping the teacher to acquire a customized and personalized user experience. More specifically:

    1. With the *Display solutions* option, the solutions of the generated exercises should also be demonstrated, along with the generated results.

2. With the *Highlight topic-sensitive words* option, an indication of possibly inappropriate words that exist in the generated exercises should be displayed, if any such word might exist. (However, it should be mentioned that only a basic python library was employed for that functionality and therefore it can only be considered as an indicative, rather than an exhaustive solution).
3. With the *Shuffle exercise instances* option, the final generated results should be shuffled in a random order, if selected. The aim of that option is to provide both of a ready-made generated exercise option, where the two verb tenses in a target teaching goal group could be evaluated in a mutual learning setting, as well as to provide a generation choice that specifically targets a single verb tense, as it was also previously mentioned. More specifically, if for example the selected teaching goal group is *Present Simple and Present Progressive* and the *Shuffle exercise instances* option is selected, SemiGramEx will generate a random mix of equal number of generated exercises both for the present simple and the present progressive verb tenses, allowing the teacher to evaluate them both in a mutual exercise. On the other hand, in a scenario where a teacher might only want to generate exercises for the present simple verb tense, the *Shuffle exercise instances* option might be remained unselected. In that way, the generated exercise instances will be demonstrated in a verb tense order, meaning that the first half of them would specifically concern present simple exercises and the other last half, would concern present progressive exercises. Such an option could allow the teacher to isolate if needed, the wanted results.

Conclusively, having selected all the aforementioned parameters, SemiGramex can generate a set of L2 grammar exercises, accordingly. To display the generated results, we have created a dedicated component that is responsible of creating a well-structured file in a *.pdf format*, where the generated results might be demonstrated. An example of such a file might be seen on Appendix C. The chosen parameters for that example, were the *Wikipedia* as an input resource, a number of *10* exercise instances, an *Advance* level of difficulty and a *FIB* type of exercise for the *Simple and Progressive Past* teaching goal group. In addition, all the three extra options were selected. On a final note, it should be mentioned that the generation procedure might take some moments, depending on the selected parameters, since for the moment it can only be executed on a local machine. In the future, the project might be hosted in a dedicated server with adequate processing power to instantly generate the target results.

Chapter 5

Evaluation

Having presented all the components of our pipeline, the final step of this work is to evaluate the overall results of the generation framework in hand. For that reason, we have separately evaluated the results of two different components of our framework, those being the *Difficulty Estimation* module and the *Generation* module. For the former, we have conducted both an automatic and a human-based evaluation, while for the latter only a human-based one. However, since the main target of our work is the human, what interest us the most in this evaluation section is the human-based assessments, while the automatic evaluation part was mostly conducted for a plurality of insights.

Nevertheless, it should be mentioned that the two human-based evaluations that were conducted, are still far from being exhaustive enough so that to derive any strong and objective claim. The limitations on time and resources that were imposed by the internship time period, had unfortunately rendered unfeasible an extensive human-based evaluation approach. Given such constraints, we were only able of employing a small number of participants to evaluate SemiGramEx, limiting in that way the objectivity of the derived results. In addition, we were restricted on demonstrating only a limited number of evaluation instances, so that to maintain the evaluation time for each of our participants on a maximum of 30 minutes (since an initial feedback we had obtained from them, had indicated for such a practice).

As a result, even though the current evaluation findings might be a good start to assess the quality of our work, they should only be carefully taken under consideration. Moreover, in an ideal scenario, besides an adequate number of evaluation participants we would also opt to evaluate our framework in a real-world educational setting. In such a setting, it could be examined on the long-term whether our framework would result an improvement of the teachers' work performance and whether it would restrict some of their most time-consuming tasks. Furthermore, in a similar manner, the suitability of our framework could also be tested on the performance and the learning experience of L2 learners.

Having reported all these, we will present on the next sections of this chapter all the results that were acquired during the evaluation procedure, and we will try to derive our conclusions based on them. We will start with the automatic evaluation that was conducted on the Difficulty Estimation module and we will conclude with the human-based evaluation that was conducted both on the Difficulty Estimation and on the Generation module.

	Precision	Recall	F1-Score	Accuracy
Beginner	0.94	0.96	0.95	
Intermediate	0.94	0.93	0.93	
Advance	0.97	0.96	0.96	
Total				**0.95**

TABLE 5.1: The scores acquired during the training of the biLSTM difficulty classifier.

5.1 Automatic evaluation of the Difficulty Estimation module

Starting with the automatic evaluation on the Difficulty Estimation module, as we have already explained on Chapter 3, we have treated the difficulty classification problem in hand, as a multi-class classification task. As a result, some of the most well-known automatic evaluation metrics for such classification tasks might be leveraged to automatically evaluate the results, such as the *Accuracy*, the *Precision*, the *Recall*, and the *F1-score* metrics.

Briefly on those metrics:

- The Accuracy metric measures the rate of correct classifications.
- The Precision metric measures the proportion of instances that turns out to be correct, in the group of instances that is declared as a class by the model.
- The Recall metric measures the proportion of instances that are correctly predicted by the model, compared to what it should actually be detected.
- The F1-score metric is the harmonic mean of Precision and Recall.

The first insights of our model's difficulty prediction strength, were obtained from the learning results during the training procedure. Based on those results, the model managed to follow a rather stable training path, reaching a final Accuracy of 95%, while also the F1-score had similar high values, for each one of the three target classes of difficulty. These scores were more than sufficient for our goal and a description of them is demonstrated on Table 5.1. In addition, as it can be seen on Figure 5.1, the training and validation loss of our model was naturally decreasing over the 7 training epochs, with the final training and validation loss of the model being adequately low and in a balanced correspondence with each other. That fact had indicated that the model did not overfit on any of the training or test set, and thus that it is able to generalize relatively good.

In addition, besides the learning results that were acquired during the training procedure, we wanted to also evaluate the results of our Difficulty Estimation module with some external, and perhaps more objective, measures. Therefore, we decided to further evaluate the generalization capability of our model, with the employement of two sentence-level L2 readability assessment corpora, where the first one of those corpora was the *OneStopEnglish* corpus (Vajjala and Lučić, 2018) which was previously presented, and the second one was the *Sentence Corpus of Remedial English* (ScoRE) (Chujo, Oghigian, and Akasegawa, 2015), a free and open-platform text resource that contains various semi-authentic sentences, written by experts to satisfy particular pedagogical conditions and annotated on three difficulty levels.

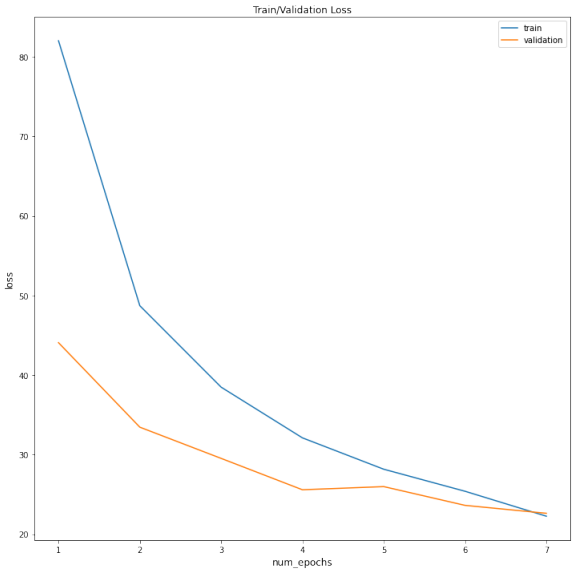


FIGURE 5.1: The training and validation loss of the difficulty classifier.

	Precision	**Recall**	**F1-Score**	**Accuracy**
Beginner	0.49	0.09	0.15	
Intermediate	0.35	0.49	0.41	
Advance	0.37	0.53	0.44	
Total				**0.37**

TABLE 5.2: The scores of the automatic classification metrics, for the OneStopEnglish corpus.

Even though both of those corpora were inadequate to be leveraged as a training resource (due to their limited extent, as it was also explained on Chapter 3), they were quite suitable to be employed as evaluation material. The results of our model on the OneStopEnglish corpus are demonstrated on Table 5.2, while for the ScoRE corpus on Table 5.3.

In a first sight, while the previous mentioned results during the model' s learning process were quite high, the generalization experiments on those two assessment corpora did not point at the same direction. More specifically, a 37% Accuracy score was obtained for the OneStopEnglish corpus and a 43% for the ScoRE corpus. In addition, the F1-score results have also demonstrated a great variability, with the macro

	Precision	**Recall**	**F1-Score**	**Accuracy**
Beginner	0.70	0.47	0.57	
Intermediate	0.23	0.35	0.28	
Advance	0.30	0.40	0.34	
Total				**0.43**

TABLE 5.3: The scores of the automatic classification metrics, for the ScoRE corpus.

average score being 33% for the OneStopEnglish corpus and 40% for the ScoRE corpus. Those results were rather poor, and in any case much lower than what we expected.

However, even though the Accuracy and the F1-score measures might be usually the desirable classification metrics, they can also be misleading in some cases. As an example of such a case for the Accuracy metric, we might consider a scenario where there is a great class-imbalance in a target dataset. In that scenario, the model might be lead to predict the value of the majority class for all the predictions and achieve in that way a high classification Accuracy score, while also being unhelpful at the same time. Similarly, in cases where a balance between Precision and Recall is not desirable, the F1-score measure might not be the best metric option as well. Therefore, for the evaluation task in hand, it might be more useful to solely examine the obtained Precision values. To explain the reason for that, we might paraphrase a little the abovementioned definitions and state that the Recall metric aims on *classifying all the possible instances of a target class* (having a higher tolerance on possible wrong classifications), while the Precision metric aims on *acquiring the best possible classification results with the less possible mistakes* (having a higher tolerance on missing possible correct classifications). Under such a perspective, a higher Precision score for our framework might be more desirable than a higher Recall score, since what we would aim for, is that the model would perform the most accurate predictions that are possible. Since a difficulty classification problem is in any case a vague task, and the way we have treated it in this work might have made it even more loose, what we primarily need is for a fast and accurate method where the teacher will not have to intensively re-examine all the results for possible mistakes (as such in the case of a high Recall score).

Nevertheless, even with a focus on the Precision metric, the model's results were not adequate. The only case where the classifier seemed to have performed properly, is in the case of the ScoRE corpus and only for the Beginner level. A precision of 70% was managed to be acquired there, which was 40% better than the results of the next best difficulty level (the case of the Advance level). Similarly, the results of the OneStopEnglish corpus for the Beginner level were also better than the next best difficulty level (Advance) in a percentage of 12%, even though these results were much lower than those reported on the ScoRE corpus, and they were in any case insufficient.

5.1.1 Analysis of the misclassifications for the two evaluation corpora

To better understand these results, we conducted a more thorough examination of the difficulty annotations that exist on these two evaluation corpora, as well as the wrong classifications that were made by our framework. However, to facilitate ourselves, we temporarily ignored the annotations and the predictions that were made for the Intermediate class (B), since that middle class might be natural to demonstrate a great mobility on the two extreme classes. As a result, we mostly examined the annotations and the predictions of the Beginner (A) and Advance (C) classes for both of our evaluation corpora, to derive the clearest possible conclusions.

Concerning the OneStopEnglish corpus, the first thing that was observed, was that in many cases, there were sentences with an extensive length, that were annotated as a Beginner difficulty level, while SemiGramEx had estimated them as an Advance difficulty level. An example of such cases might be seen on Table 5.4 for three sentence examples. On the other hand, many of the Advance level annotated sentences of the OneStopEnglish corpus, were rather short in terms of sentence length

and they were probably annotated based on the appearance of specific, advanced-level words, while SemiGramEx had predicted those sentences as a Beginner level. Such examples are illustrated on Table 5.5. Eventually, another fuzzy point that was observed on the annotations of the OneStopEnglish corpus, concerned the way that these annotations were created. More precisely, it appeared that in many cases the same sentence would have been annotated with different difficulty levels, while only changing some words or small parts of it, each time. That way of annotating sentences rendered the distinction between different difficulty annotations a rather subtle task even for us, and it justified in some extent the wrong classifications of the model. An example of two tuples of such sentences' annotations, are demonstrated on Table 5.6 and Table 5.7.

Model Prediction	**OneStopEnglish Label**	**Sentence**
C	A	British and Dutch researchers develop new form of lie-detector test Ewen MacAskill, defense and security correspondent January, 2015 Police and intelligence agencies around the world have, for almost 100 years, used the polygraph, a lie-detector test, to help catch criminals and spies.
C	A	Many of the world's billionaires might agree with this way of thinking but it would be a very big change for most workers and their employers.
C	A	"The Commission will introduce a new emissions test that will properly check the cars in real driving", said Lucia Caudet, a Commission spokesperson.

TABLE 5.4: Sentences annotated with beginner difficulty level on the OneStopEnglish corpus and the model's predictions.

Model Prediction	**OneStopEnglish Label**	**Sentence**
A	C	People send us pictures of them, framed and laminated.
A	C	All these people on the pier were staring down at me open-mouthed.
A	C	No project is deemed too wacky.

TABLE 5.5: Sentences annotated with advance difficulty level on the OneStopEnglish corpus and the model's predictions.

On the other hand, the sentence annotations of the ScoRE corpus appeared to be much more straightforward. In this corpus, most of the existing sentences were relatively restrained in terms of sentence length and the difficulty annotations appeared to be much more intuitive to us, than those of the OneStopEnglish corpus. However, in this corpus too, there were many cases of fuzzy annotations, such as sentences that were labeled as a Beginner label by SemiGramEx (a label that was

Model Prediction	True Label	Sentence
B	A	Statistics suggest that smokers and recent ex-smokers (the majority of vapers) may already be using e-cigarettes less.
C	B	Statistics suggest that vaping among smokers and recent ex-smokers, who are the vast majority of vapers, may already be declining.

TABLE 5.6: An example of two sentences from the OneStopEnglish corpus, annotated with different difficulty levels, while only changing some small parts of them.

Model Prediction	True Label	Sentence
C	A	The vulnerable northern white rhino has nearly been hunted to extinction in spite of the guards and their guns.
C	B	The vulnerable northern white rhino has been hunted very nearly to extinction in spite of every precaution, in spite of the guards and their guns.

TABLE 5.7: Another example of two sentences from the OneStopEnglish corpus, annotated with different difficulty levels, while only changing some small parts of them.

also in compliance with our intuition), which had been initially annotated as an Advance level on the corpus. Indicatively, two examples of such sentences' annotations can be seen on Table 5.8.

Model Prediction	True Label	Sentence
A	C	The American School in Japan is usually called ASIJ.
A	C	It has been so great to see you tonight at the school reunion.

TABLE 5.8: Two sentences from the ScoRE corpus, annotated with the advance difficulty level and the model's predictions.

5.1.2 Conclusion on the automatic evaluation part

To conclude with this evaluation part, what we have seen until now in both cases of the previous mentioned evaluation corpora, is that many possible issues might pre-exist on the way that those corpora were created and annotated. More specifically, we saw that many of the sentence annotations existing on the OneStopEnglish corpus might be unintuitive and with frequent discrepancies. As a characteristic example of such a case, we have seen that quite frequently sentences with an extensive sentence length might be annotated with a Beginner difficulty level and sentences with a limited sentence length might be annotated with an Advance difficulty label. On the other hand, the ScoRE corpus comes with a more controlled length of

sentences and the difficulty annotations seem to be mostly related with the lexical difficulty of each sentence. However, even in that case many annotation discrepancies were also identified. Therefore, all these points might strongly indicate that the annotations of those two corpora cannot be considered as a gold-standard by any means.

In fact, having only those first insights from the automatic evaluation part, the only conclusion that we were able to conduct, is that the difficulty estimation ability of our model might be in a tight correspondence with the length of the target sentence. That would perhaps explain why the only adequate results that were obtained, were the Precision score on the Beginner level for the ScoRE corpus. Given that the sentences in the ScoRE corpus, had maintained a relatively short sentence length and the difficulty annotations were oriented on the lexical complexity, our model had no great difficulties on properly classifying such short sentences of a Beginner level. In a similar sense, it was unable to perform well on sentences of the Advance level, since for such a level the model might primarily expect sentences of a greater length.

As a result, we might state that the insights we have obtained from this automatic evaluation part, are still not enough to clearly decide about the true quality of our framework. More precisely, due to the diversity that was demonstrated on all these automatic evaluation results, what is mainly being proved in our opinion, is that such objective measures might not be the most suitable way to evaluate the constantly changing nature of a difficulty estimation task. On the contrary, a subjective, human-based evaluation might appear to be a much more valid approach to assess the task in hand, and thus, that might be the evaluation approach that we might trust the most. Between the high result scores obtained from the training procedure and the lower result scores obtained from the two aforementioned evaluation corpora, the true quality of our model might lie somewhere in the middle and a human-based evaluation might hopefully play a clarifying role on this.

5.2 Human-based evaluation

5.2.1 Evaluation of the Difficulty Estimation module

For the human-based evaluation part of the Difficulty Estimation module, we have asked 4 evaluators to answer a short evaluation form. All the evaluators were L2 teachers with a high English language adequacy. The teachers were separated in two groups, where for each group 15 sentences were demonstrated and they were asked to estimate which difficulty level, from the pre-existing 3 we have presented, they believed was the most suitable for each sentence (we will subsequently refer to this evaluation part as the *first evaluation section*). In the evaluation form, general information about the nature of our work were provided and the teachers were asked to perform their annotations in a manner, similar to what they would do during their duties in a L2 educational setting (meaning that no strict guidelines were given, but rather the intuitive method that they would follow on such a task, was demanded). The purpose of that evaluation section was to study the correspondence between the difficulty estimations of L2 teachers and the estimations of our model, for a given set of sentences. All the sentences that were demonstrated in this evaluation part, were randomly retrieved from our Wikipedia corpus.

In addition, we further asked our evaluators to order a set of sentences based on their difficulty hierarchy. More specifically, 6 sentence triples were demonstrated on each group of teachers. Each of those triples contained three sentences presented

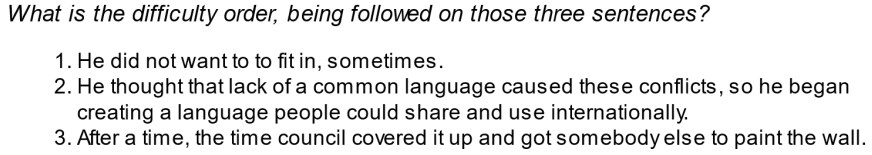


FIGURE 5.2: An evaluation example for a difficulty hierarchy triple, where a possible answer might be 1:A, 2:C, 3:B.

in a random order, with each sentence corresponding to one of the three reported difficulty labels. Each teacher had to provide an estimated difficulty order that corresponds to the order that the sentences of the triple were presented. Then, the difficulty order that was chosen by each teacher was compared with the difficulty order that was chosen by the model, to understand in which extent those two converge. In Figure 5.2, an example of such a sentence triple can be seen (we will subsequently refer to this evaluation part as the *second evaluation section*). The aim of that evaluation section, was to examine the relative difficulty estimation of the model, meaning to see whether the distinction between different difficulty levels, that the model was able to do was meaningful, leaving aside whether the predictions themselves were accurate enough. Similarly with the first evaluation section, no strict guidelines were provided to the evaluators for that task, but rather it was asked from them to follow an intuitive approach, similarly with what they would do in a real-world teaching setting. All the sentences that were demonstrated in this part, were also randomly retrieved from our Wikipedia corpus.

5.2.2 Evaluation of the Generation module

Besides assessing the quality of the Difficulty Estimation module, we have also conducted a human-based evaluation for the final exercises that are produced by the Generation module. For that part of the evaluation, the evaluators were the same 4 teachers, like the human-based evaluation part of the Difficulty Estimation module, with similar evaluation guidelines. However, in this evaluation part, the teachers were asked to evaluate a set of final generated exercise instances. More specifically, for each supported exercise type (Fill-in-the-blank, Multiple-choice and Find-the-mistake), there were demonstrated 5 exercise instances to the teachers, resulting a total of 15 exercise instances per teacher. Moreover, the difficulty level and the target teaching goal for those exercise instances were chosen randomly since time limitations had restricted us on simultaneously evaluating also those aspects. Eventually, all those exercise instances were retrieved from the Wikipedia input resource (we ignored the BNC corpus since only an experimental version is supported for the moment, and a reproducibility of this evaluation part could be easily conducted in the future when a stable version of it might be available).

In Figure 5.3, the 4 evaluation options that were demonstrated for each exercise instance can be seen (we will subsequently refer to this evaluation part as the *third evaluation section*). Each one of those options represented an evaluation dimension of a corresponding exercise instance, with those being the *grammaticality* of our exercises (i.e., the degree that an exercise instance was grammatically correct), the *ambiguity* of our exercises (i.e., the degree of context-dependence and the clarity in terms of meaning), the *appropriateness* of our exercises (i.e., the degree that an exercise instance was appropriate in terms of the target teaching goal), and the *preparedness* of

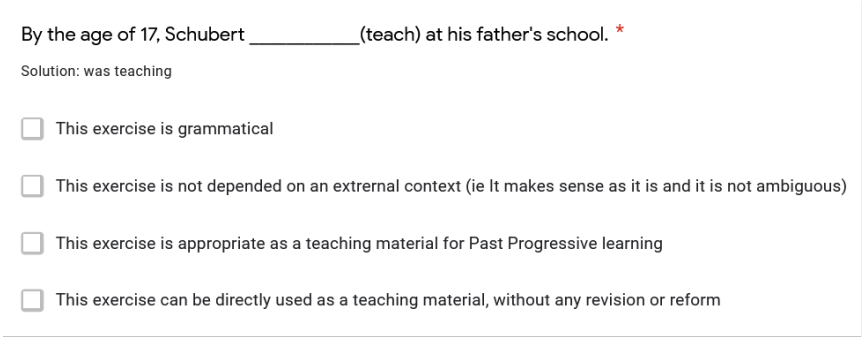


FIGURE 5.3: The 4 evaluation dimensions that were demonstrated during human-based the evaluation of the final generated exercise instances.

our exercises to be directly incorporated in a L2 setting (without an extensive revision). Each exercise instance could be subject to one or more of those 4 dimensions. However, out of the four dimensions, we were less interested on the *appropriateness* dimension, since the limited extent of demonstrated evaluation questions had restricted us on exhaustively evaluate all the provided teaching goals. That dimension was mostly examined indicatively, so that possible discrepancies on that direction might be revealed. Having conducted all these, we were able of evaluating the quality of SemiGramEx' s generated outcomes as educational material to be used in a L2 setting. Similarly with the Difficulty Estimation evaluation part, the exercise instances presented in this part were randomly selected, so that any possible bias would be eliminated.

5.2.3 Results of the two human-based evaluations

5.2.3.1 First and second evaluation section

Concerning the first evaluation section, only 53% of the human classifications in total, were similar with the classifications that were made by SemiGramEx. However, in 77% of those cases, the framework's difficulty prediction had coincided with at least one of the teachers' decisions, indicating that the labels assigned by SemiGramEx were not completely irrelevant with the human estimations (we remind at this point that each sentence was estimated by two teachers belonging to one of the two evaluation groups). On the other hand, for the second evaluation section, the results were much more adequate. Around 67% of the human classifications in total were the same with SemiGramEx' s difficulty estimations and in 92% of those cases, there was at least one similar prediction of SemiGramEx with one of the teachers belonging to the group. Therefore, the first insights on these results indicated that, while the framework's difficulty predictions had some distance with the human predictions (even though they were not completely irrelevant), the way that the framework was ordering triples of sentences in a difficulty hierarchy was in an adequate correspondence with the human estimations. Therefore, it could be stated that, even though SemiGramEx was not able of estimating the difficulty of a sentence in the most accurate way, the estimations it was making (in comparison with each other) were intuitive enough.

However, to further stretch the credibility of the above conclusion, we have also measured the *Inter-Annotator Agreement* (IAA) among the teachers. In that way, we could more accurately estimate the objectivity of the human estimations, since a

Evaluation section ID	Evaluation group ID	IAA
1	1	0.50
1	2	0.56
2	1	0.78
2	2	0.40
Avg		**0.56**

TABLE 5.9: The IAA scores for the first two evaluation sections.

coiling of the human annotators against the predictions of SemiGramEx, would indicate a lesser validity for the framework's estimations. *Cohen's kappa* was chosen as the most suitable IAA measure since it is a quite frequent reliability measure of such purposes, for two raters who rate the same thing, that also takes under consideration a possible chance agreement. A complete annotator agreement for that measure corresponds to the value of 1 and the value of 0 is equivalent to chance agreement. A score that is less than 0 indicates that there is no agreement at all. The results for both sections and evaluation groups, are demonstrated on Table 5.9.

While for each section and group, the IAA values are much higher than the threshold of chance agreement, they are still inadequate to derive robust conclusions. According to the threshold range proposed by McHugh, 2012, almost all the retrieved IAA scores are in the range of 0.41-0.60, which accords to only moderate annotator agreement. The only exception of this is the IAA score of the first group for the second evaluation section and therefore, all that can be derived from these scores is that the difficulty estimation task in hand seems to be rather subjective and contradictory, even for professional teachers.

5.2.3.2 Third evaluation section

Subsequently, concerning the third evaluation section, the results of it are demonstrated on Table 5.10. In that Table, it can be seen that the four evaluated dimensions, had generally managed to acquire high scores. More specifically, almost all the demonstrated sentences were considered as grammatical, and a 78% percentage of the exercises were evaluated as appropriate teaching material for the given verb tense. (Even though, as it was already mentioned, that dimension could not be exhaustive, the score that was acquired is an optimistic first indication of that framework's aspect.) Furthermore, the demonstrated exercises were evaluated as adequate to be directly used as a teaching material without any revision or reform, in a 78% percentage too. Given that the aim of this framework is primarily to aid L2 teachers and that we had assumed from the beginning, that a revision of the generation results would be almost mandatory, the percentage that is reported on this dimension is utterly encouraging, indicating that the framework might already be in a much better status than what we had initially thought of.

The only low result in this evaluation section was the one concerning the context-dependence/ambiguity dimension. In that dimension, SemiGramEx had only managed to acquire a 52% score, indicating that the *Sentence Selection* component we had incorporated (thoroughly presented on Subsection 2.2.3), was not sufficient to adequately overcome the context-dependence issue of our input resources. Nevertheless, the fact that under these circumstances a high rate of evaluators had accepted the demonstrated exercises as a teaching material with no need of revision (with a score of 78%), was a rather intriguing fact. Based on the low score of the context-dependency dimension, we would expect that most of the sentences would have

Evaluated dimension	Score in total
This exercise is grammatical.	97%
This exercise is not depended on an external context (i.e., It makes sense as it is, and it is not ambiguous).	52%
This exercise is appropriate as a teaching material for VERB-TENSE (the VERB-TENSE corresponds to the verb tense that is evaluated used each time).	78%
This exercise can be directly used as a teaching material, without any revision or reform.	78%

TABLE 5.10: The results of the third evaluation section.

been evaluated as ambiguous by the teachers, and thus necessary to be thoroughly revised. However, an extra qualitative analysis that was performed on the retrieved results, had granted us with a perspective that explained this intriguing issue. More specifically, in each evaluation section, we had also provided further commenting options so that the teachers could elaborate more on their evaluations if wanted. Therefore, as it was derived through those comments, the teachers were aware of the contradiction that was emerging with the evaluation of those two different dimensions. However, they believed that even if many of the exercises were initially context-depended and ambiguous, in the way they were demonstrated during the evaluation, it might be possible that a detailed exercise instruction, provided by the teacher, would make those exercises suitable to be incorporated in a L2 setting without any further revision. As a result, even though one of the major flaws of our approach was not able to be entirely solved yet, it was indicated that the framework might still be useful as it is.

5.2.3.3 Qualitative comments for all the evaluation sections

Eventually, before we conclude this section, we might also elaborate a little more on the rest of the comments that were retrieved during the human evaluation.

Based on those comments:

1. The vocabulary that was incorporated on the first evaluation section for the Beginner difficulty level, was considered a bit challenging. Even though it was clarified from the beginning of that evaluation task, that the Beginner difficulty level might not correspond to absolute beginners (but rather to learners with some initial language background), it seems that still the vocabulary in use might be harder than expected in some cases. A possible reason for that, might lie on the encyclopedic nature of our Wikipedia input resource (the sentences of that evaluation part were retrieved from that input resource), since lots of proper names and formal words might exist there.

2. Some teachers had indicated that while a distinction between the Beginner and the Advance difficulty level was quite easy to be performed, it was much harder for them to estimate between the Beginner-Intermediate and the Intermediate-Advance levels. Such indications might further underline the subtle nature of that middle category, even for a human.

3. For the second evaluation section, most of the teachers had reported that it was much easier to estimate the difficulty hierarchy rather than estimating the difficulty level of an isolated sentence (the case of the first evaluation section). Such comments might strongly indicate again, the subjective nature of a difficulty estimation task. In that way, the decision of a text passage's difficulty level might be rather difficult as a standalone task, while the estimation of the difficulty of the same text passage, in relation with another one, might be a much more straightforward task.

4. Most of the teachers reported that sentence length was a decisive factor during the difficulty estimation procedure they had conducted, both for the first and for the second evaluation section. That comment is in correspondence with the behavior we have previously reported for our framework, concerning the importance of the sentence length on the framework's estimations.

5. For the third evaluation section, most of the teachers believed that even though there might be some discrepancies on the final generated exercises (some exercises might have ambiguous meaning or an ambiguous use of a verb tense), those exercises would most likely not need an extensive revision, meaning that small changes, such as the insert of an adverb or the change of a noun, might render them suitable for a L2 teaching setting. That comment is in correspondance with the conclusion we arrived, about our framework being already in an adequate state.

6. All the evaluators had estimated that SemiGramEx was generally rather useful as a support on their duties, and all of them had reported a great user experience when tested the framework's web-interface.

Chapter 6

Conclusion and future work

In this thesis, we have developed and presented a framework that generates English L2 grammar exercises for verb tenses learning. A collection of English sentences were leveraged as a generation basis for that purpose, which were annotated with a state-of-the-art parser with different layers of linguistic information. In that way, the framework can perform a set of queries on that collection, to retrieve sentences based on a target teaching goal, before converting them into appropriate grammar exercises. Rule-based techniques and handcrafted patterns were incorporated for that purpose and 3 types of exercises are supported, with those being the Fill-in-the-blank, the Multiple-choice, and the Find-the-mistake types of exercise. In addition, a difficulty classification component was developed, so that the generated grammar exercises might also be relevant to a target L2 proficiency level. For that reason, we have incorporated a DL model, trained on a custom training corpus we created, that accepts English sentences as an input and results a difficulty class, from a predetermined difficulty range, as an output. Eventually, a set of extra options that aim on better personalizing the user-experience, were also incorporated.

The final evaluation results we have retrieved, indicated that the framework might already be adequate, in general terms, to be a helpful and valuable tool for L2 English teachers. More specifically, we have separately evaluated the results of the difficulty classification component and the final generated exercises. For the former, the results we have obtained from various automatic evaluation measures had been varying from utterly good to relatively poor. Therefore, to clarify the true quality of this component we have further conducted a human-based evaluation, based on which, the task in hand appeared to be greatly vague even for a human. More specifically, it appeared that while the difficulty estimations that were performed by the framework were only in a medium correspondence with the estimations that were performed by the evaluators, an extensive disagreement was also demonstrated between the decisions of the evaluators, underlining in that way the subjective nature of that task. Eventually, a human-based evaluation was also conducted on the final generated exercises, where more than adequate results were reported on almost all the evaluated dimensions.

Conclusively, before we finish this work, there exist some minor and some major threads of improvement, that we believe they should be ameliorated in the future.

Starting with the minor improvements:

- Different input resources could be incorporated as a generation basis to extend the supported teaching goals, since for the moment only 6 verb tenses are supported, which are those that mainly exist on the input resources in use.
- An extra input resource option could be employed, where the teacher could insert an input text of preference as a generation basis. Such an option might be very important to further personalize the framework's generation results.

For this to happen, an initial issue that should be surpassed might be to decide about the text form that would be accepted in such a case (i.e., a limited input text field, or a complete file in .pdf format, etc.). Furthermore, another issue might be to develop ways that such an input text could be pre-processed, since the pre-processing techniques we have incorporated in this work might not result the best possible outcomes on generic-purpose text passages (all of our pre-processing steps, were created in a "trial and error" spirit, specifically for the text resources in hand). Eventually, another major issue might be to develop novel ways that such an input text could be parsed with the relevant linguistic information in the fastest possible way, since the incorporation of a state-of-the-art parser like the one leveraged in this work, might add a great amount of time on the generation procedure.

- Only three difficulty classes are supported by SemiGramEx for the moment, since our initial experiments had indicated that a 3-label classification approach would result more distinct classes and robust estimations. However, we have decided to follow that path mainly due to the limited extent of the graded lexicon that was leveraged in this work as an annotation basis, since for that reason, the difficulty categories of the lexicon were packed togethere into more generic ones, in order to acquire categories with higher number of contained instances. If a more extensive lexicon would be retrieved, it might also be possible to incorporate a more extensive range of difficulty classes.

- The only provided way for the moment, that a teacher could isolate a specific teaching goal (i.e., a specific verb tense), is with the *Shuffle exercise instances option*, as it was thoroughly explained on Chapter 4. With that option, the generated results can be demonstrated in a mixed group or in two distinctive ordered groups, where each group corresponds to each verb tense in a teaching goal group. However, in that way there still exist some manual user effort and therefore, a dedicated solution to isolate the teaching goals could be employed.

- The *Highlight topic-sensitive words* option is only incorporating, for the moment, a standard python library to perform a profanity check and thus, it cannot be considered as an exhaustive solution by any means. Further research might improve the results of it, perhaps with the employment of dedicated topic modeling techniques.

Eventually, concerning the major improvements, the most important issue that should be resolved in our estimation, concerns the ambiguity that derives from the context dependence of the sentences that are used as a generation basis on SemiGramEx. That deficiency mainly derives from the way that our input resources have been leveraged, since those were split into sets of their sentences before being converted into exercise instances. In that way, the final exercise instances appear to be frequently insufficient as standalone language examples, due to the extensive dependencies they might initially had maintained with their surrounding context. Among the employed types of exercises, that context-dependence deficiency has mostly demonstrated a great effect on specific types of exercises. More precisely, while the Fill-in-the-blank type of exercise has proven to be already sufficient in terms of such a quality, the Find-the-mistake and Multiple-choice types, might still suffer in a great extent from an ambiguity of that sort. That is expected, since in many of those cases the surrounding context might be decisive on the choice of the right option (i.e., due to the absence of context, all the solution options might appear

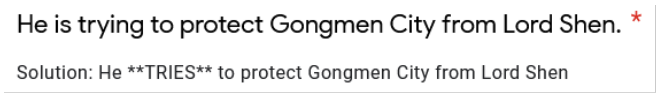


FIGURE 6.1: An ambiguous Find-the-mistake exercise, where both of the incorrect (Present Progressive) and the correct (Present Simple) tenses could be correct, depending on the surrounding context.

to be suitable). An example of such an issue for a Find-the-mistake exercise, can be seen on Figure 6.1.

To resolve that issue, a sentence selection module was employed, which was extensively presented on Subsection 2.2.3. However, the conditions we had applied on that implementation step, have proven to be only partly sufficient, since the evaluation of the framework's final generation results indicated that there is still much room for improvement. Therefore, perhaps a more extensive incorporation of the filtering choices that were reported on Pilán, Volodina, and Borin, 2017, might be a solution.

On the other hand, the second major deficiency of SemiGramEx, is that it is strictly referred to the English language for the time being. SemiGramEx should eventually evolve into a language-independent generation framework, or at least a multilingual one in case the language-independent goal is ultra-optimistic. In fact, the initial objective of this work was to implement a generation framework that would also function on other languages besides English. However, the rule-based generation techniques we have leveraged, appeared to be much more restrictive for such a task, than what was initially thought of. More specifically, we have searched for ways that those rule-based techniques might be "translated" into corresponding generation rules for other languages, but we were not able to obtain any valuable result that might be possible to be implemented on this internship timeframe. Nevertheless, there is always the possibility that a more extensive experimentation and research, could render the grammar generation task in hand more language independent.

Conclusively, besides all these major and minor improvements, the final goal of this framework is to also incorporate other aspects of language learning, such as vocabulary and comprehension exercises. DL techniques could probably be leveraged for the generation of them, instead of the rule-based methods that were incorporated in this work. In addition, another extension of this project would also be to acquire a user-side that would specifically concern L2 learners. In that way, this framework might become a complete educational toolkit that can aid both teachers and learners in a L2 language learning setting. Ideally, those two project sides might be able to interact in a way that the performance of a learner might trigger relevant generation outcomes for the teacher.

Appendix A

A snapshot of the CEFRLex graded lexicon

Lemma	POS-tag	A1	A2	B1	B2	C1	Total
cat	NN	77.40	351.71	39.19	28.57	22.53	79.38
empty	JJ	0	28.83	28.65	102.29	37.84	61.88
explore	VB	0	153.38	60.50	109.99	205.43	130.37

FIGURE A.1: The CEFRLex graded lexicon.

Appendix B

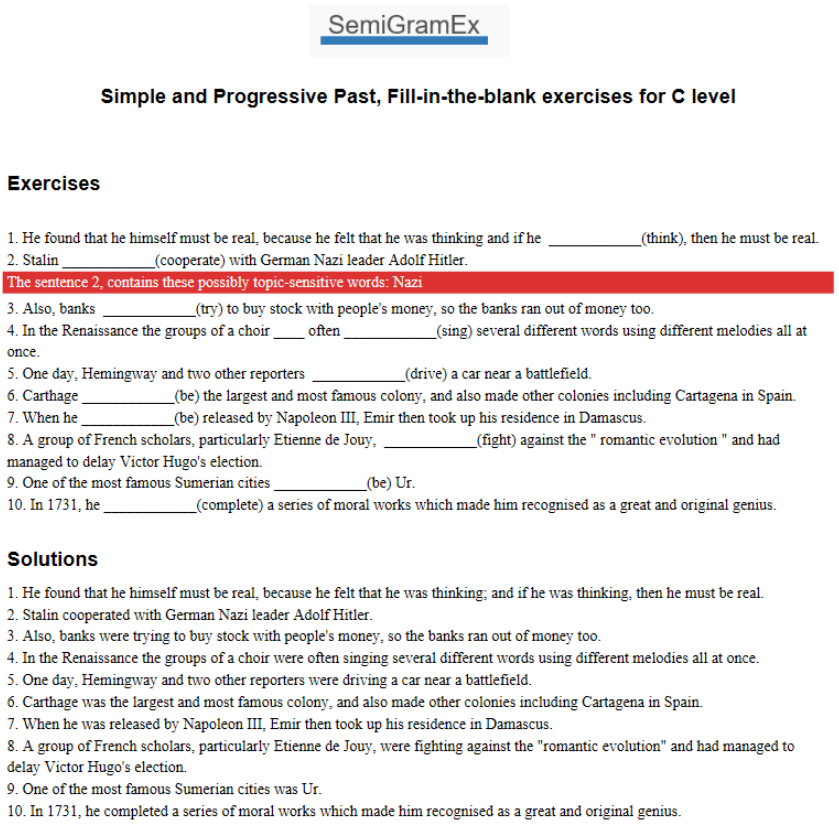
The main supporting parameters of SemiGramEx

Corpora	**Difficulty levels**	**Types of exercises**	**Verb tenses**	**Teaching goals**
Simple Wikipedia	Beginner	Fill-in-the-blank	Present Simple	Present Simple and Progressive
BNC	Intermediate	Multiple-choice	Present Progressive	Past Simple and Progressive
	Advance	Find-the-mistake	Past Simple	Present and Past Perfect
			Past Progressive	
			Present Perfect	
			Past Perfect	

TABLE B.1: Overview of the main parameters of SemiGramEx.

Appendix C

The generation results in a .pdf format



SemiGramEx

Simple and Progressive Past, Fill-in-the-blank exercises for C level

Exercises

1. He found that he himself must be real, because he felt that he was thinking and if he _____(think), then he must be real.
2. Stalin _____(cooperate) with German Nazi leader Adolf Hitler.

The sentence 2, contains these possibly topic-sensitive words: Nazi

3. Also, banks _____(try) to buy stock with people's money, so the banks ran out of money too.
4. In the Renaissance the groups of a choir ____ often _____(sing) several different words using different melodies all at once.
5. One day, Hemingway and two other reporters _____(drive) a car near a battlefield.
6. Carthage _____(be) the largest and most famous colony, and also made other colonies including Cartagena in Spain.
7. When he _____(be) released by Napoleon III, Emir then took up his residence in Damascus.
8. A group of French scholars, particularly Etienne de Jouy, _____(fight) against the " romantic evolution " and had managed to delay Victor Hugo's election.
9. One of the most famous Sumerian cities _____(be) Ur.
10. In 1731, he _____(complete) a series of moral works which made him recognised as a great and original genius.

Solutions

1. He found that he himself must be real, because he felt that he was thinking; and if he was thinking, then he must be real.
2. Stalin cooperated with German Nazi leader Adolf Hitler.
3. Also, banks were trying to buy stock with people's money, so the banks ran out of money too.
4. In the Renaissance the groups of a choir were often singing several different words using different melodies all at once.
5. One day, Hemingway and two other reporters were driving a car near a battlefield.
6. Carthage was the largest and most famous colony, and also made other colonies including Cartagena in Spain.
7. When he was released by Napoleon III, Emir then took up his residence in Damascus.
8. A group of French scholars, particularly Etienne de Jouy, were fighting against the "romantic evolution" and had managed to delay Victor Hugo's election.
9. One of the most famous Sumerian cities was Ur.
10. In 1731, he completed a series of moral works which made him recognised as a great and original genius.

FIGURE C.1: A set of FIB exercises generated by SemiGramEx in a .pdf format, for the simple and progressive past tenses and the advance difficulty level.

Bibliography

Agirrezabal, Manex et al. (2018). "Creating vocabulary exercises through NLP". In: *Digital Humanities in the Nordic Countries. Proceedings, 2019*.

Alsop, Sian and Hilary Nesi (2009). "Issues in the development of the British Academic Written English (BAWE) corpus". In: *Corpora* 4.1, pp. 71–83.

Beinborn, Lisa, Torsten Zesch, and Iryna Gurevych (2012). "Towards fine-grained readability measures for self-directed language learning". In: *Electronic Conference Proceedings*. Vol. 80, pp. 11–19.

Chalvin, Antoine, Egle Eensoo, and François Stuck (2013). "Mining a parallel corpus for automatic generation of Estonian grammar ex-ercises". In: *Electronic lexicography in the 21st century: thinking outside the pa-per: proceedings of the eLex 2013 conference, 17-19 October 2013, Tallinn, Esto-nia*, pp. 280–295.

Chen, Chia-Yin, Hsien-Chin Liou, and Jason S Chang (2006). "Fast–an automatic generation system for grammar tests". In: *Proceedings of the COLING/ACL 2006 Interactive Presentation Sessions*, pp. 1–4.

Chujo, Kiyomi, Kathryn Oghigian, and Shiro Akasegawa (2015). "A corpus and grammatical browsing system for remedial EFL learners". In: *Multiple affordances of language corpora for data-driven learning*, pp. 109–130.

Collins-Thompson, Kevyn and James P Callan (2004). "A language modeling approach to predicting reading difficulty". In: *Proceedings of the human language technology conference of the North American chapter of the association for computational linguistics: HLT-NAACL 2004*, pp. 193–200.

Dale, Edgar and Jeanne S Chall (1948). "A formula for predicting readability: Instructions". In: *Educational research bulletin*, pp. 37–54.

Deutsch, Tovly, Masoud Jasbi, and Stuart Shieber (2020). "Linguistic features for readability assessment". In: *arXiv preprint arXiv:2006.00377*.

Dürlich, Luise and Thomas François (2018). "EFLLex: A graded lexical resource for learners of English as a foreign language". In: *Proceedings of the Eleventh International Conference on Language Re-sources and Evaluation (LREC 2018)*.

Fenogenova, Alena and Elizaveta Kuzmenko (2016). "Automatic generation of lexical exercises". In: *Proceedings of the International Conference*.

Flesch, Rudolph (1948). "A new readability yardstick." In: *Journal of applied psychology* 32.3, p. 221.

Forti, Luciana et al. (2019). "Measuring text complexity for Italian as a second language learning pur-poses". In: *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pp. 360–368.

Heilman, Michael and Noah A Smith (2010). "Good question! statistical ranking for question generation". In: *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 609–617.

Heilman, Michael et al. (2007). "Combining lexical and grammatical features to improve readability measures for first and second language texts". In: *Human language technologies 2007: the conference of the North American chapter of the association for computational linguistics; proceedings of the main conference*, pp. 460–467.

Hoshino, Ayako and Hiroshi Nakagawa (2005). "A real-time multiple-choice question generation for language testing: a preliminary study". In: *Proceedings of the second workshop on Building Educational Applica-tions Using NLP*, pp. 17–20.

Hubbard, Phil (2012). "Curation for systemization of authentic content for autonomous learning". In: *EUROCALL Conference, Gothenburg, Sweden*, pp. 22–25.

Hwang, William et al. (2015). "Aligning sentences from standard wikipedia to simple wikipedia". In: *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 211–217.

Lee, John SY and Mengqi Luo (2016). "Personalized exercises for preposition learning". In: *Proceedings of ACL-2016 System Demonstrations*, pp. 115–120.

Leech, Geoffrey Neil (1992). "100 million words of English: the British National Corpus (BNC)". In:

Lo Bosco, G, Giovanni Pilato, and Daniele Schicchi (2019). "A recurrent deep neural network model to measure sentence complexity for the italian language". In: *AIC 2018, Artificial Intelligence and Cognition 2018*. Vol. 2418, pp. 90–97.

Marrese-Taylor, Edison et al. (2018). "Learning to automatically generate fill-in-the-blank quizzes". In: *arXiv preprint arXiv:1806.04524*.

Martinc, Matej, Senja Pollak, and Marko Robnik-Šikonja (2021). "Supervised and unsupervised neural approaches to text readability". In: *Computational Linguistics* 47.1, pp. 141–179.

McHugh, Mary L (2012). "Interrater reliability: the kappa statistic". In: *Biochemia medica* 22.3, pp. 276–282.

Mitkov, Ruslan et al. (2003). "Computer-aided generation of multiple-choice tests". In: *Proceedings of the HLT-NAACL 03 workshop on Building educational ap-plications using natural language processing*, pp. 17–22.

Modern Languages Division, Council of Europe. Council for Cultural Co-operation. Education Commit-tee. (2001). *Common European Framework of Reference for Languages: learning, teaching, assessment*. Cambridge University Press.

O'Connor, Rollanda E et al. (2002). "Teaching reading to poor readers in the intermediate grades: A comparison of text difficulty." In: *Journal of Educational Psychology* 94.3, p. 474.

Pennington, Jeffrey, Richard Socher, and Christopher D. Manning (2014). "GloVe: Global Vectors for Word Representation". In: *Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1532–1543. URL: http://www.aclweb.org/anthology/D14-1162.

Perez-Beltrachini, Laura, Claire Gardent, and German Kruszewski (2012). "Generating grammar exercises". In: *The 7th Workshop on Innovative Use of NLP for Building Educational Ap-plications, NAACL-HLT Worskhop 2012*, pp. 147–157.

Pilán, Ildikó, Sowmya Vajjala, and Elena Volodina (2016). "A readable read: Automatic assessment of language learning materials based on linguistic complexity". In: *arXiv preprint arXiv:1603.08868*.

Pilán, Ildikó, Elena Volodina, and Lars Borin (2017). "Candidate sentence selection for language learning exercises: from a com-prehensive framework to an empirical evaluation". In: *arXiv preprint arXiv:1706.03530*.

Qi, Peng et al. (2020). "Stanza: A Python natural language processing toolkit for many human lan-guages". In: *arXiv preprint arXiv:2003.07082*.

Schwarm, Sarah E and Mari Ostendorf (2005). "Reading level assessment using support vector machines and statistical language models". In: *Proceedings of the 43rd Annual Meeting of the Association for Computa-tional Linguistics (ACL'05)*, pp. 523–530.

Si, Luo and Jamie Callan (2001). "A Statistical Model for Scientific Readability". In: *Proceedings of the Tenth International Conference on Information and Knowledge Management*. CIKM '01. Atlanta, Georgia, USA: Association for Computing Machinery, 574–576. ISBN: 1581134363. DOI: 10.1145/502585.502695. URL: https://doi.org/10.1145/502585.502695.

Skory, Adam and Maxine Eskenazi (Sept. 2010). "Predicting Cloze Task Quality for Vocabulary Training". In:

Soler, Aina Garí, Marianna Apidianaki, and Alexandre Allauzen (2018). "A comparative study of word embeddings and other features for lexical com-plexity detection in French". In: *Actes de la Conférence TALN. Volume 1-Articles longs, articles courts de TALN*, pp. 499–508.

Soonklang, Tasanawan and Weenawadee Muangon (2017). "Automatic question generation system for English exercise for secondary students". In: *the 25th international conference on computers in education*.

Vajjala, Sowmya and Ivana Lučić (2018). "OneStopEnglish corpus: A new corpus for automatic readability assessment and text simplification". In: *Proceedings of the thirteenth workshop on innovative use of NLP for building educational applications*, pp. 297–304.

Vajjala, Sowmya and Detmar Meurers (2012). "On improving the accuracy of readability classification using insights from second language acquisition". In: *Proceedings of the seventh workshop on building educational applica-tions using NLP*, pp. 163–173.

— (2014). "Assessing the relative reading level of sentence pairs for text simplification". In: *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 288–297.

Xia, Menglin, Ekaterina Kochmar, and Ted Briscoe (2019). "Text readability assessment for second language learners". In: *arXiv preprint arXiv:1906.07580*.

Zhu, Zhemin, Delphine Bernhard, and Iryna Gurevych (2010). "A monolingual tree-based translation model for sentence simplification". In: *Proceedings of the 23rd International Conference on Computational Lin-guistics (Coling 2010)*, pp. 1353–1361.