UNIVERSITÉ DE LORRAINE

MASTER THESIS

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

Author:
Chrysovalantis MASTORAS

Supervisor:
Prof. Yannick PARMENTIER

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

in the

Synalp team
IDMC

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

Loria
IDMC

Master of Natural Language Processing

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

by Chrysovalantis MASTORAS

In this work, we present a framework that can generate Second Language (L2) exercises in a semi-automatic manner, namely *SemiGramEx*. More precisely, given a set of input parameters such as a target teaching goal and a difficulty level, SemiGramEx can generate a set of L2 grammar exercise instances. The framework supports for the moment verb tenses learning, with 6 verb tenses being leveraged, three different types of exercises and three 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.

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 tasks might be the retrieval and the manual creation of appropriate 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.

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's degree, as well as my fellow 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.

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

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

Chapter 1

Introduction

1.1 Project overview

Since the 1960s many research efforts have been reported, trying to integrate the scientific domain of Natural Language Processing (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 acquisition by reducing human intervention with the help 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 to reclaim such available NL data and facilitate the language acquisition process with the creation of appropriate educational content. For such a purpose, many of the most typical and well-known NLP modules can be proven 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 will be 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. The target of this work is L2 learning, and the goal is to enhance, if possible, such a learning procedure. More specifically, in this work, we will present the implementation of a generation system for L2 exercises that combines DL and rule-based methods. 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 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 teaching goal that our system targets 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 extend this work on other languages also exist. Even though most of the existing research on the subject has been targeting L2 learners, our work is focused on the aid of L2 teachers by automating the searching and construction procedure of the educational content, a task that is quite frequent

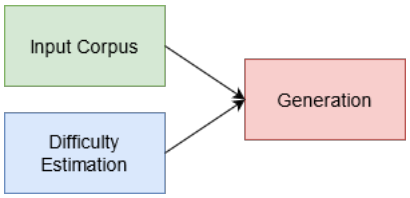


FIGURE 1.1: The three modules of SemiGramEx.

on the duties of a L2 teacher. More specifically, given a set of input parameters by the teacher, the system's goal is to instantly generate concrete grammar exercises to be used in a L2 teaching setup. In that sense, the generation framework in hand is a semi-automatic one, since the L2 teacher can be considered as an active and indispensable component of the generation procedure.

SemiGramEx is consisted of three major modules. An *Input Corpus* module, a *Difficulty Estimation* module and eventually a *Generation* module. The generation flow 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 annotated with different layers of linguistic information that is 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 teaching setup. For that reason, a DL classifier has been incorporated where given an input sentence, the 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 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. Given an additional difficulty level by the teacher, those sentences are eventually filtered correspondingly. The 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 automating the tedious task of creating conjugation exercises?

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

1. How to retrieve from a given text, a set of sentences that are adequate for a target L2 learner (in terms of a 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 1, we will shortly present the working environment where the internship and

this thesis were conducted, along with the equipment and the tools that were used during our work. Then, on Chapter 2 we will introduce the first module of our pipeline, the Input Corpus. An overview of relevant examples from the recent literature will be presented, as well as the input resource 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's 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. Its goal is to develop tools for learners and teachers, taking under consideration the learners' profiles by the means of learning analytics. 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, as well as the incorporated tools and 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 specifically, the existing research departments at LORIA are the *Algorithms, Computation, Image and Geometry* department, which focuses on geometry and symbolic

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

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 their research interests are focused on hybrid, symbolic and statistical natural language processing approaches. Their research topics 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

Our work required the implementation of a pipeline of different modules, where each one of them will be thoroughly elaborated subsequently. 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 (as such NLTK[7], ScikitLearn[8], Pandas[9], Spacy[10], etc.). In addition, a web-interface of the overall implementation was also created, mainly for demonstrative purposes. For that reason, the Flask[11] framework was incorporated, as well as the HTML5, CSS3 and JQuery[12] programming languages.

The internship 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. Google colab[13] was leveraged as the main coding environment, even though the Jupyter Notebook[14] interface and the PyCharm[15] framework, were also employed in some extent, mostly for tasks that were mandatory to be implemented locally. 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 all the computationally intensive tasks. Secondly, google colab is a much easier setup when it comes to the installation of various libraries and their dependencies, simplifying time-consuming procedures of

[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/

that kind in a great extent. During the production stage of our implementation, a google drive[16] workspace was created where each 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, where it 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 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 Gitlab[17] repository was created where the code of this work is shared, and the reader can also have the chance to examine many details of this project in the corresponding README.md file.

Conclusively, during this internship period, a set of difficulties have been 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 work colleagues, appeared to be a rather challenging setting in this work. Even though the frequent meetings with the internship supervisor had greatly improved this 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-organizing and autonomicity, were rapidly developed. In addition, another challenge that was confronted, had to 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 initially conducted, to build a feasible research topic proposal. That task was perhaps the most challenging one that had to be confronted, mainly due to inadequate experience on such procedures. Nevertheless, that challenge was also probably the most educative one that was able to be acquired during this internship period, since it gave the chance of confronting a real-word research example from the very first step of it to the very last, rendering this internship period a holistic research experience.

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

[17] https://gitlab.com/valadisprf94/semi-automatic-generation-of-grammar-exercises

Chapter 2

Input Corpus

The first component of our work is the *Input Corpus* module that was also previously mentioned, where 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 exercises' generation. The first approach is to incorporate an input resource as a generation basis and the other is to directly generate exercises with the help of automatic text generation techniques. Even though both approaches are actively present on the recent literature, the automatic text generation approaches had mostly been adopted for comprehension or vocabulary types of exercises. On the contrary, most of the proposed generation approaches for the task of automatic generation of grammar exercises, in our knowledge, had incorporated an 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 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 also thoroughly report all the decisions and the steps that were made during the implementation of the Input Corpus module. We will avoid on presenting an overview of literature examples concerning the comprehension and vocabulary types of exercises that were mentioned, since that would overstretch the aim of this chapter. We will only be restricted on literature examples that concern grammar exercises and we will continue this chapter under the assumption, that the incorporation of an input resource as a generation basis is indeed a valid approach for the task in hand. However, in the subsequent Chapter 4, some of the most indicative generation methods in the relevant literature will also be reported, and 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 an 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 [1], where pedagogically oriented and well-formed text material can be obtained, before being converted into the appropriate exercise form. Subsequently, the second resource class that was reported mostly 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 [2] and [3], who based their generation on documents and sentences retrieved from the Web, or [4] and [5] 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, [6] had used as a generation basis two major corpora namely the *British Academic Written English Corpus* [7] and the *British National Corpus* [8], 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 having been 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, as such in the cases of [9], [10] and [11] accordingly. Eventually, even though much less frequently, parallel corpora have also been incorporated for a specific target language, as in the example of [12].

As it was illustrated, a great diversity of leveraged input resources has 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. In fact, a long-held debate exists in the L2 research community concerning that issue.

The two main types of language usages on L2 exercises that have mainly been occupied that discussion, are those of *synthetic language examples* and *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. Elaborating more on those two categories, the synthetic-language examples have been in fact a widely used case of language learning material. They 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 come without any manual effort, since with that approach the abundance of free text on the Web can be incorporated. 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 has been strongly criticized when used in a L2 setting since many infrequent words might be contained. More importantly, the very notion of authentic language use might have a rather different meaning for

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

First Language (L1) and L2 learners, where such an input resource might be suitable for the former but inappropriate for the latter.

While each of those viewpoints appear to have sufficiently convincing arguments, we had eventually 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. More specifically, we have decided to treat the generation process of our task, in a computationally rather than a pedagogically oriented way. 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 input material is an aspect of great importance, it will only consist of a secondary concern for us. (However, it should be mentioned that this perspective concerns only the exercises generation process, 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 and the overall quality of our framework, pedagogical validity maintains the most important role). That decision was taken mostly 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 for our research since even if any possible discrepancies, in terms of pedagogical validity, might exist in our input resource, the teacher will always have the chance to evaluate the final generation results and possibly reform them in an appropriate manner. In addition, the research of [13] at that point had also empowered our position. In this work, it was argued that authentic language materials can be a proper material for a language learning setting if the approach is not entirely automatic, but rather a human proofreading and editing 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 resource used on SemiGramEx

To implement the Input Corpus module, three additional sub-components were developed, 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 extracted from the collected input resource and enriched 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 that collection would remain. An illustration of all those components is demonstrated in Figure 2.1 and each one of them, will be subsequently thoroughly presented.

2.2.1 Scraping

The first component of the Input Corpus, namely Scraping, was responsible of collecting the appropriate text material and converting it into the appropriate form.

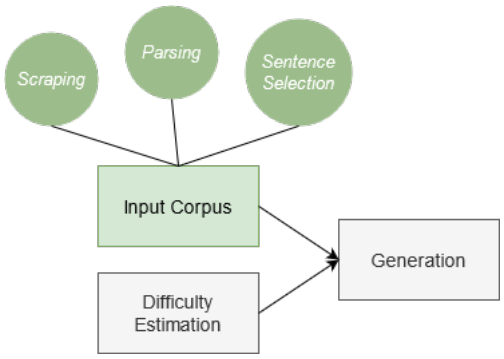


FIGURE 2.1: The components of the Input Corpus module.

Having decided that we will incorporate an input resource of authentic language text 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 would be freely available, easily accessible, adequate in terms of language quality and size length and that it would demonstrate an authentic language use. In addition, it should retain in a minimum extent some formal 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 setting of L2 learning. 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. 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 second language 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 input resource. Those articles were retrieved with the Wikipedia API[4] through the list of *vital articles of level 4*[5] that Wikipedia offers. The Wikipedia vital articles are mainly collections of article lists that are published by Wikipedia and concern a diverse range of subjects that every Wikipedia database should have. After being scraped, those articles were pre-processed and cleaned to maintain only the main content of each article and the most relevant information. All the pre-processing steps were designed specifically for the form of a Simple Wikipedia article. Subsequently, each of those scraped articles were split into a collection of sentences. Several further cleaning and processing steps were applied to those sentences and some of the most common abbreviations were expanded (e.g., the abbreviation *I'm* would be expanded to *I am*), since that would be more helpful for the python libraries employed during that step

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

of our pipeline. 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 of 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, where there may exist some possible discrepancies 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.

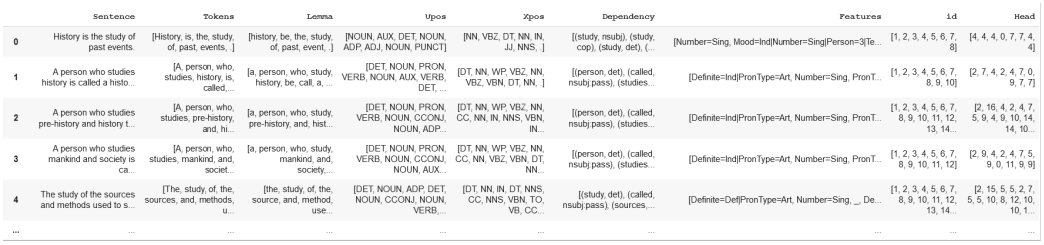
To create that secondary input resource, a BNC subset of 4M words namely the *small BNC corpus*[6] was incorporated. In a similar manner with the one that was described for the Simple Wikipedia corpus, several pre-processing steps were performed, which were precisely designed for the text form of the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

2.2.2 Parsing

Having acquired the two sentences' collections, the next step was to retrieve the additional linguistic information of those sentences. In that way, a set of queries could be performed on those input resource collections, 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 incorporated the *Stanza parser* [14] 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 with a multi-lingual support.

The parsing procedure was similar for both the input collections. 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 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

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



	Sentence	Tokens	Lemma	Upos	Xpos	Dependency	Features	id	Head
0	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, .]	[(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]
1	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, .]	[(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]
2	A person who studies pre-history and history t...	[A, person, who, studies, pre-history, and, hi...	[a, person, who, study, pre-history, and, hist...	[DET, NOUN, PRON, VERB, NOUN, CCONJ, NOUN, ADP...	[DT, NN, WP, VBZ, NN, CC, NN, IN, NNS, VBN, IN...	[(person, det), (called, nsubj:pass), (studies...	[Definite=Ind|PronType=Art, Number=Sing, PronT...	[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14...	[2, 16, 4, 2, 4, 7, 5, 9, 4, 9, 10, 14, 14, 10...
3	A person who studies mankind and society is ca...	[A, person, who, studies, mankind, and, societ...	[a, person, who, study, mankind, and, society,...	[DET, NOUN, PRON, VERB, NOUN, CCONJ, NOUN, AUX...	[DT, NN, WP, VBZ, NN, CC, NN, VBZ, VBN, DT, NN...	[(person, det), (called, nsubj:pass), (studies...	[Definite=Ind|PronType=Art, Number=Sing, PronT...	[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12]	[2, 9, 4, 2, 4, 7, 5, 9, 0, 11, 9, 9]
4	The study of the sources and methods used to s...	[The, study, of, the, sources, and, methods, u...	[the, study, of, the, source, and, method, use...	[DET, NOUN, ADP, DET, NOUN, CCONJ, NOUN, VERB,...	[DT, NN, IN, DT, NNS, CC, NNS, VBN, TO, VB, CC...	[(study, det), (called, nsubj:pass), (sources,...	[Definite=Def|PronType=Art, Number=Sing, _, De...	[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14...	[2, 15, 5, 5, 2, 7, 5, 5, 10, 8, 12, 10, 10, 1...
...	...	...	...	...	...	...	...	...	...

FIGURE 2.2: The Simple Wikipedia dataframe after parsing.

dataframe contained 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, 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 might be incorporated on the next steps of our pipeline. Having acquired all that linguistic information, the system was at that point able to retrieve appropriate sentences based on a set of target linguistic constructions, before converting them into the appropriate final exercise form. The preprocessing steps that were performed had cleaned the sentences from any irrelevant information, while the parsing steps had revealed the hidden linguistic information. Nevertheless, those sentences were still far from being adequate for our task. Given that those sentences 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 a standalone L2 exercise. 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 [15] to apply a set of better sentence selection rules and filter out the two corpora.

2.2.3 Sentence Selection

In [15] 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 consisted of 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 [15] was Swedish, a fact that imposed many restrictions to 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 mostly 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. In addition, *structural well-formedness* of a sentence 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 can 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.

Subsequently, several criteria were further leveraged to cope with *context-dependence*. Syntactic aspects of context-dependence were examined based on various *structural connectives* (i.e., *than*). The leveraged connectives were manually retrieved from various linguistic resources, and thus they cannot be considered as extensive. With the incorporation of such structural connectives, one can identify paradigms of coordinating or subordinating conjunction, were 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 [15], 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.

Having acquired the aforementioned criteria, they were all 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 small 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. In the next chapter, we will present the second component of our pipeline which is responsible on classifying the existing corpora, based on a language proficiency criterion.

Chapter 3

Difficulty Estimation

3.1 Relevant literature

3.1.1 Readability assessment in general

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, with their 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 learner's proficiency level, as [16] 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. However, classifying educational content based on their difficulty has proven to be a demanding objective, rather than a straightforward task. Such tasks usually try to tackle with questions of the sort: *which are those factors that consist of the difficulty of a given educational material?* and they must be treated as a multifaceted problem to be solved.

Typically, the task of difficulty classification of educational content might be seen under different perspectives. The type of a given L2 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. In addition, difficulty might also be seen under a strictly pedagogical perspective, where for example an educational material is classified based on specific pedagogical conditions. The Common European Framework of Reference for Languages (CEFR) [17] grading is one of the most famous examples of such a difficulty scale, where a range of learner proficiency levels along with a set of pedagogical guidelines about the prerequisite knowledge that a learner should acquire in each of those levels, has been presented. As a result, the procedure of classifying an educational material under such a perspective is usually the work of a group of experts, who are trying to utilize their domain-specific knowledge to properly classify an educational content based on the pedagogical constants of the CEFR.

However, for our task, we have decided to incorporate an approach that is oriented on the difficulty of the plain language itself rather than any other pedagogically oriented aspect. One important factor that dictated us to follow that path, similarly with the decisions we had made during the Input Corpus 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 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. All these had led us into the subject of *Readability Assessment*, a field with an extensive research history, based on which we have decided to confront our task.

Readability assessment has been a subject with long history that firstly originates back to the 1920s and aims on properly identifying the appropriate 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 text is based on. Due to the importance of such a difficulty estimation method, there have been many efforts 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.

Starting from classic readability formulas and the incorporation of surface-based text features to state-of-the-art Machine Learning (ML) approaches, the field has been largely evolved during the last recent years, from rather simplistic approaches to complex ones, following all the major trends of the NLP domain. In addition, DL approaches have also been introduced, even though in a much smaller extent, indicating that the field is still alive and kicking. Various examples from the major trends of those approaches, will be more thoroughly presented in the following part of this chapter. We will start from traditional readability approaches, and we will end with the state-of-the-art ones. Furthermore, we will also introduce some of the most frequently referenced readability corpora, an inseparable component of any modern readability assessment procedure. Even though we will eventually devote our interest on readability assessment that specifically targets L2 language learning settings, the aim of that retrospection is to guide the reader through an overview of the involvement of this field, as well as the different methods that have been employed so far. Through that overview, the limitations and all the existing problems of this topic should be revealed, and the main challenges that we had to confront during the implementation of this part should also be indicated.

3.1.2 Different approaches of readability assessment

Traditionally, the task of readability assessment had been tackled with the use of various readability assessment formulas. Perhaps the [18] and [19] are 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, targeting L1 readers. The [18] formula was initially created to help the U.S army control the difficulty of mainstream newspapers and it was intended for adults. The syntactic feature that is used for prediction is the average number of words per sentence, while the lexical one is the average number of syllables per 100 words. On the other hand, [19]'s formula, was targeting primary and secondary grade school children. The syntactic feature that is used is also the average number of words per sentence, while the lexical one is the number of words that do not exist in a predefined list, including 3000 lexemes. Nevertheless, even though these formulas had managed to provide some valuable results and they are still extensively applied on various domains, they have been strongly criticized for their simplicity since they only consider superficial features, ignoring other possible important aspects of readability (such as text coherence etc.).

Many different approaches had been subsequently introduced to surpass the inconsistencies appeared on the traditional formulas, as for example the method of [20] 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 and retrieving eventually results that were much more accurate than the ones of [18]' s formula. In addition, [21] had tackled the readability task in a similar setting a little later, with a language model using a Multinomial Naïve Bayes classifier. On the other hand, one of the first attempts of a Support Vector Machines (SVM) classifier model for that subject, had been introduced in the approach reported by [22]. In that work, a diverse combination of features was incorporated, using statistical LMs, along with other traditional and surface-based features. Nevertheless, most of those approaches at that time had been targeting the readability of L1 readers, an approach that might not be the best practice for L2 learning setting, since many differences may exist on the way that L1 and L2 reading acquisition happens, as it was also indicated by [23].

It was not until the 2007 when the field of readability assessment changed its orientation and L2 reader targets also started to be extensively examined. A first example of that is the work of [24], who reported a much greater importance of the grammatical features of their work for L2 readability, in comparison with L1 readability. Subsequently [25], had also incorporated developmental Second Language Acquisition (SLA) measures in their research and combined them with traditional readability features such as word and sentence length. A superior importance of lexical features compared to syntactic ones for L2 learners reported in their results, indicating that L2 readability assessment might be subject to different difficulty factors than those concerning a L1 setting. In addition, perhaps one of the most important approaches on this subject during the last recent years, would be the state-of-the-art approach for L2 readability of [26]. In that paper, a SVM classifier was incorporated along with a diverse set of training features (such as traditional surface-level features, parse tree syntactic features, language modeling features etc.) and furthermore, an experiment was conducted on ways to resolve the issue of limited available L2 data, by incorporating L1 readability data. Eventually, besides being L2 specific, readability assessment had also been subject on the specific language being used. Starting with the work of [27] who had incorporated a SVM algorithm to tackle L2 readability assessment using language-independent features, numerous L2 readability assessment approaches have been introduced for multilingual settings with other languages besides English. Indicatively, in [28] the collection process of a French corpus that is intended to be used for readability assessment, is being thoroughly described. Furthermore, in [29] and [30], ML classifiers have been employed based on various linguistic features to predict the difficulty level of Swedish and Italian L2 text, accordingly.

Besides being L2 specific, another perspective of readability assessment that should be taken under consideration, is the structure-level of it. More specifically, most of the approaches 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 a field that requires much more research. Assessing for a sentence-level readability model might prove to be a much more demanding task than what one might initially consider, as it was also confirmed by [31], who only managed to acquire an accuracy of 71% for a binary sentence classification task. Similarly, [29] only 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. Eventually, the low accuracy results at the sentence-level assessment task of

[32], in comparison with the adequate accuracy scores they obtained for document-level tasks, should also be indicative points that sentence-level readability research has still much progress to do.

Eventually, one last research approach that has only recently occurred in the field of readability assessment, is that of DL methodologies. During all the previously mentioned literature, we have seen an incorporation of a diverse range of readability assessment approaches. Starting from traditional readability assessment formulas that used superficial prediction features, we have seen that the field had been revolutionized with the prevalence of ML algorithms which managed to overcome many of the previous deficiencies. Nevertheless, such algorithms usually come with the burden of an appropriate feature engineering approach to perform the best possible results, demanding for a set of fine-grained and aspect-oriented linguistic features that should be manually defined. These features, as it was also indicated before, can be rather simple ones (such as the sentence length or the character count) or even much more sophisticated (such as information from syntactical parse trees or semantic features) and they are 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 eventually appears to be greatly depended on them and the results will only be as good as the acquired features. To overcome that issue, 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 expertise and instead, they only require for a large amount of quality training data and a rather efficient model complexity, to resolve a 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, [33] presented a Recurrent Neural Network (RNN) model that automatically detects whether an Italian sentence is simple or complex. [34] on the other hand, leveraged much more complex architectures and managed to achieve state-of-the-art performance with a pretrained Transformer model and a Hierarchical Attention Network (HAN) model, two complex and efficient DL architectures, in two well-known readability corpora. Furthermore, [35] had also reported promising results by exploring the role of linguistic features in DL methods and indicating that there exists no improvement at all with the incorporation of them, signifying in that way, that these DL models might already be able to implicitly capture the features that are useful for readability assessment. All those research findings are shaping an optimistic future for the field of readability assessment, connotating that this subject is scientifically active and perhaps near in the future, even much better results might occur. Nevertheless, in our knowledge, the research findings on DL readability assessment for the moment are much more obscure when it comes to readability assessment for L2 educational contexts. Most of the existing L2 readability assessment approaches restrict themselves to traditional or ML methodologies, leaving an open research space on ways that DL models could also be incorporated for L2 readability assessment. In addition, such models as we already mentioned are intensively data-depended on and in that way, a great necessity on freely available and extensive L2 readability assessment corpora, emerges.

One of the first examples of such L2 readability assessment corpora with an extensive size, was that of [22]. That corpus, namely the *WeeklyReader corpus*, was a

document-based collection made to enhance L1 and L2 teachers during their teaching duties. For that reason, different versions of an educational newspaper, corresponding to four different grade levels, were incorporated. The collection procedure had targeted children for different spans of ages and eventually, it contained a variety of around 2,400 grade-annotated articles of non-fiction topics. A few years later, [25] had further extended the WeeklyReader corpus, presenting a new readability assessment corpus that was dubbed as the *WeeBit corpus*. To create that corpus, the WeeklyReader corpus was combined with material received from the BBC Bitesize[1], an educational website targeting children of specific age spans and graded into four difficulty levels. The combined WeeBit corpus, was assembled using a broad range of classes that intended to readers with ages between 7 and 16. In the end, the result was a corpus of around 3,100 annotated documents. Eventually, a more recent corpus of readability assessment was recently proposed by [26], namely the *Newsela corpus*. Having as a starting point the absence of adequate L2 readability assessment corpora, the authors of the Newsela corpus had collected a set of 1,900 news articles for L2 learners. Each article contained several simplifications, targeting in that way readers at different reading levels. As a result, each article in the corpus had been re-written for up to 4 times, providing in that way a range of grade levels.

However, even though the aforementioned corpora might be rather useful options for many L2 readability assessment tasks, they come the burden of being annotated on a document or a paragraph level. As a result, for situations of settings were sentence-level annotated corpora are necessary, they might prove to be rather unsuitable. 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 [36] that document-level assessment methods would be unreliable for shorter texts, such as single sentences. In that way, a great necessity exists for readability corpora that are exclusively annotated on a sentence level.

A scarce number of research approaches for that matter have been reported. Most of these approaches had tried to generate a corpus with the leverage of the Wikipedia and the Simple Wikipedia databases. The Simple Wikipedia database, is a version of Wikipedia where writers had been requested to create their article entries in simple words and short sentences, targeting mostly children or adults learning the target language. Therefore, a sentence-alignment procedure could be leveraged between the sentences of these two databases. Such sentence-aligned corpora have mostly been used for text simplification tasks, but also for binary readability assessment tasks in the sentence-level, where the Simple Wikipedia sentences are considered as easier targets in comparison to the Wikipedia ones and labels corresponding to that distinction are created in a custom way. Among many existing approaches on creating such aligned corpora, we might mention the approaches used in [37] and [38] indicatively. In the former, a corpus of around 100k aligned Simple Wikipedia and Wikipedia sentence pairs was created, where the alignment between two sentences was only conducted if their similarity score was greater than a given threshold. On the other hand, in the latter paper, a greedy search method was incorporated to acquire the relevant candidate sentences from the two databases and then a word-level semantic similarity score that was based on the Wiktionary[2] was leveraged, capable of accounting both for semantic and syntactic similarities.

[1] https://www.bbc.co.uk/bitesizel
[2] https://www.wiktionary.org/

Eventually, besides incorporating resources such as the Wikipedia and the Simple Wikipedia, the *OneStopEnglish* [39] corpus was another solution of recently proposed L2 corpus, annotated at the sentence-level. 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 levels of adult L2 learners, those being elementary, intermediate, and advanced, resulting in that way a manually annotated L2 document-level readability assessment corpus. Subsequently, the corpus was converted from the document-level into the sentence-level. To do so, the authors had incorporated the cosine similarity measure, and they performed a one-to-all comparison for all the texts, taking each time one pair of their reading levels. In the end, the final corpus that was acquired, was a sentence-aligned corpus of 6,994 sentence pairs.

3.1.3 Conclusion on readability assessment

In conclusion, it appears that the task of readability assessment is a much more delicate subject, than what one might initially thought of. We have seen that the difficulty of a given text can be determined from different perspectives, starting from linguistic oriented features to tightly pedagogical ones. In addition, a plethora of available approaches have been reported over the past recent years, from simple readability formulas to state-of-the-art ML approaches. Again there, readability assessment appears to be subject on various and subtle factors, such as the features to be incorporated or even the target audience and the structure of the training corpus in use. Concerning specifically the topic of L2 readability assessment, a set of even more challenging and hardly spotted factors, might further exist. The style of writing or the text format, the learner's native language and cultural background, the age, and the level of conceptual familiarity, might only be some of the many variables that could be decisive when assessing the readability of a text passage for L2 educational purposes. As a result, the readability assessment of a target text appears to be a rather vague and subjective task that is greatly depended on the quality of the selected criteria, based on which the procedure happens.

Given all that, in our work we have decided to employ the advantages of DL techniques to overcome on the biggest possible extent, the necessity of intensively specifying such necessary readability assessment criteria. 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 text sentence in an appropriate difficulty label. However, these DL techniques, as we have mentioned, are tightly depended on the available training data, based on which any necessary dimension for the task in hand, could be inferred. Therefore, to overcome the absence of adequate L2 training corpora annotated on a sentence-level, we have further created our own custom training corpus. 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 grade lexicon. Having those two resources, each one of those sentences were automatically annotated based on the entries of the grade 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

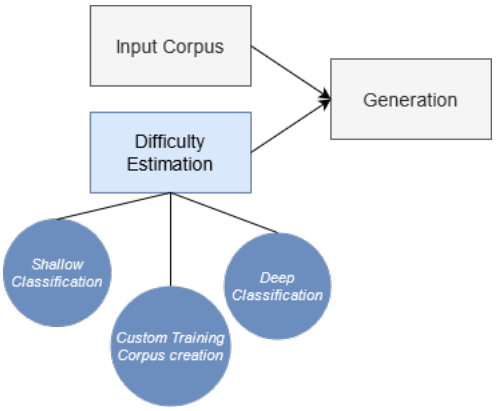


FIGURE 3.1: The components of the Difficulty Estimation module.

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 our task, was the absence of sufficient sentence-level annotated training corpora. That issue was perhaps the most important that we had to confront during our implementation, since the ability of a DL model to resolve complex problems, as it was already explained, is tightly related with the quality and the amount of training data. Initially, we had conducted numerous experiments on some of the previously reported sentence-level and publicly available corpora. Nevertheless, those available corpora are still far from being adequate for a modern DL model mostly due to their limited extent and therefore, the results we had managed to acquire for our purpose were rather poor. On the other hand, we were persistent not to follow a ML approach for that task (a decision that might have solved the limited training corpus size), since that would require in some extent a necessity of some expert-knowledge for the feature engineering process and furthermore, evaluating and selecting combinations of relevant features would most likely proven to be a rather time consuming and resource-intense task. Our goal was to overcome all those inconsistencies with a DL approach and for that reason, we decided to create our own training corpus. However, manually annotating a corpus is, similarly with a ML approach if not in a much greater extent, a rather time consuming and expert-knowledge-depended process. Therefore, we aimed on manually 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 [20], where in an automatic generation system for preposition exercises, 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 of, the intuition that the vocabulary of a sentence (based on a predefined manually graded lexicon, rather than any specific and linguistically oriented lexical aspect) could demonstrate some sort of difficulty classification validity, had been the starting point of our thinking on that subject. More precisely, we have initially followed a similar way to create the custom annotated training corpus in 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 implement such a Shallow Classification, we had initially leveraged the *CEFRLex graded lexicon* of [40] 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. While several other possible resources might had served for that task, such as one of existing and well-known word frequency lists, those 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 distribution presented on the lexicon, to the assignment of a specific difficulty label on each word, we followed an approach similar with [41]. More specifically, we assigned to each word of the lexicon, the CEFR label of its first appearance, meaning that a word would be considered on a given CEFR label if that CEFR level would be the first time of its appearance, indicating in that way the moment of acquisition. Moreover, to make the categories as distinct and robust as possible, we clustered the 5-scale CEFR labels, 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 retrieving 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 convert those 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 wanted to avoid the conversion of a document into sentences, mostly to avoid any possible discrepancies that would 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 in use on any such corpus should be generic and of every-day use, to avoid any bias on incorporated topics or the target language form.

Given all that, we were tightly restricted on avoiding several possible sources of well-known English sentences' collections. For example, one option might had been

to incorporate any of the existing English parallel corpora that are usually incorporated in machine translation tasks, where the English sentences could be isolated from their translations and form in that way a set of collected sentences. However, these parallel corpora are usually constructed for a specific topic or with the use of a very formal type of language, hiding a possible classification bias in that way for our model. On the other hand, another option might had been to randomly scrape the Web for possible resources of our need. Nevertheless, that option carried the danger of collecting a resource that contained an uncontrollable use of language, since it is quite usual that even some typical language formalities might not be kept in such settings. It was our initial estimation after all, that a collection of English sentences written in a language that would obey in some sort of writing principles, should be a much more valid option.

Eventually, a last point that was important on 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 words 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 words without an entry in our lexicon. 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, and problems as the aforementioned, would not exist.

To the best of our knowledge, the publicly available English sentence-level corpora that fulfill the above criteria are not abundant and therefore, small compromises had to be done in terms of quality of the data. As a result, we had selected as the best candidate for that task the set of English sentences provided by the *Tatoeba database* [3], since we observed a rather optimal behavior of it with our model, in comparison with other candidate corpora sources that we tried.

The Tatoeba project, is self-introduced as a collection of sentences and translations that are 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, where anybody can propose new sentences 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. 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 provided by Tatoeba. In that list, the proficiency level of each translator is reported and therefore, we filtered the collected English sentences only on 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 the collection of English sentences from the Tatoeba database,

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

as well as the CEFRLex difficulty-grade lexicon, the next step in our implementation would be 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 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 a 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 since those were the main part-of-speech tags existing in our grade 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 corresponding with the four mentioned part-of-speech tags, had to simultaneously exist in our grade 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 on it as a second, *Deep Classification* step. While we could only follow an approach similar with [20] and simply stick on classifying the difficulty based on the sentences' vocabulary, we were restricted on doing so, since the grade lexicon we incorporated was quite limited. On the contrary, our intuition was that if we would provide to the DL model a shallow categorization as an initial guidance (in our case based on the grade lexicon), then the model might be able to learn on its own all the rest, underlying difficulty factors that might render it capable to generalize on unknown sentences. 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 creation process of our custom training corpus is demonstrated in Figure 3.2.

Concerning the DL model architecture that we employed for the task in hand, that was a bidirectional Long Short-Term Memory (biLSTM) one. We found that this architecture was sufficient to capture the non-linear features' dependencies and the 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 model. We initially experimented with different layer sizes and other architecture features or hyperparameters before we conclude to the most adequate candidates. All the hyperparameters presented, 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 [42] were incorporated, to encode the 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.

During the creation of our custom training corpus, we balanced our three classes to 69,900 sentence instances for each, since above that limit we noticed 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

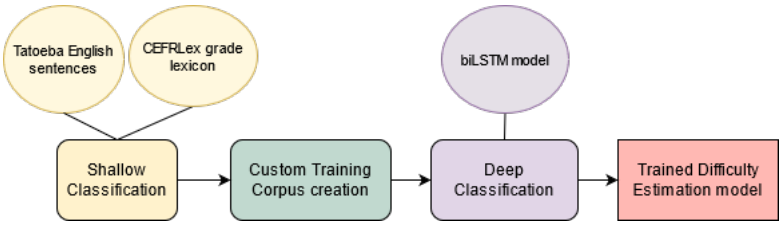


FIGURE 3.2: The two-fold classification process on the custom training corpus.

with a further evaluation of our model's results, will be subsequently presented on Chapter 5, where an evaluation of our work will be reported.

Chapter 4

Generation

Both of the components that have been reported until now, they had been 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 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 difficulty level and a teaching goal, among others. In that way, a L2 teacher might have the chance of eliminating some of the most time-consuming tasks of a L2 teaching procedure and to primarily focus on the important teaching activities.

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. Subsequently, we will also present the details of all the implementation steps we have performed to create the Generation module. Eventually, the overall SemiGramEx framework interface will 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 [3], who introduced the *FAST* generation system. That framework had incorporated a semi-automatic generation method for grammar exercises, based on text material retrieved from the Web. A set of handcrafted patterns and rules were leveraged to transform that text material into appropriate grammar exercises. More specifically, the FAST system was able of retrieving sentences from the Web and examining whether they would match with a set of predefined target rules. In such a case, the sentences would be converted into Multiple-Choice and Error Detection types of questions. More precisely, based on a set of predefined generation patterns, the retrieved sentences were converted into having erroneous statements mixed with correct ones as demonstrated possible solutions or wrong statements in the place of the original correct ones, accordingly.

Subsequently, a more recent approach on that subject might be the approach proposed by [1]. In that paper, 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 sentences, annotated with relevant linguistic information, would be acquired. Having acquired such annotated sentences, only the relevant sentences of them were targeted based on specific grammatical goals and they were subsequently converted to a given type of an exercise with a rule-based approach. A diverse set of language learning goals were

included in this work, such as *adverbs learning*, *prepositions learning*, and *verb tenses learning*. Furthermore, four different types of generated questions were supported. Those were the classic Fill-in-the-blank exercises, Either/Or exercises (where a correct answer between two choices must be chosen), True or False questions (where the learner must also correct the false question) and Error Correction (where the incorrect word in a sentence must be spotted).

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 the one of [12] for Estonian, where French-speaking learners were targeted. In that example, an Estonian-French parallel corpus had been incorporated as a generation basis, along with a custom difficulty classification approach and a set of content selection criteria. More specifically, the system was able to retrieve relevant sentences from the existing parallel corpus and convert them into appropriate Fill-in-the-blank grammar exercises, by replacing the target grammatical form with the corresponding lemma. The types of exercises that were supported by the system aimed on practicing skills concerning the morphology of words (by constructing the relevant form of a target grammar construction), or the syntactical competence (by choosing the appropriate syntactic form of a word in a context).

However, even though the approaches that were presented so far have been extensively incorporating an 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 [43], namely *GramEx*. In that approach, instead of employing a predefined corpus as a generation basis, the authors had created a system that was able to directly generate grammar exercises in a controllable manner, concerning both aspects of lexical and syntactical complexity of a sentence. To do that, a formal grammar was leveraged, where the syntax and the semantics of the allowed generated sentences were described and, in that way, the system was able to generate a set of exercises targeting a specific pedagogical goal. In more details, typical surface realization techniques were initially used to construct a set of candidate sentences. Then, the most relevant generated sentences were retrieved based on a predefined constraint language. Having acquired such sentences, those were then converted into appropriate grammar exercises. 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, where the function words of the sentence were deleted, and the remaining lemmas were shuffled before being demonstrated to the user.

Eventually, a research work that should be presented before we conclude this overview is the one that was recently proposed by [10], which is also the most similar one with our approach. More specifically, [10] 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. However, unlike with our approach, the approach reported in this work had been targeting a L2 learner user rather than a L2 teacher. As a result, a dedicated module was further incorporated, so that the system would be able to adapt the generation process based on a learner's behavior. In a similar manner, for each exercise, a set of distractors were also generated based on statistics retrieved from various learner corpora.

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 consist of an input resource that is being processed, so that appropriate text materials can be retrieved, which in their turn, are being converted into appropriate grammar exercises. However, as we have already reported on Chapter 2, a more extensive elaboration on the relevant literature would reveal to the reader that this is not the case for comprehension or vocabulary L2 exercises. On the contrary, the most likely adopted approach nowadays for such examples, might be to treat the generation task as a typical *Automatic Text Generation* task, with the incorporation of a DL technique (employing typical sequence-to-sequence architectures etc.). That fact, had initially imposed a great challenge on our research, concerning whether we should also follow a DL approach for the task in hand. However, the frequency that an input resource was incorporated on most of the research examples reported before, had eventually convinced us to incorporate an input resource as a generation basis as well. In fact, in our estimation, the main reason that typical ML or DL approaches had not been extensively incorporated on generation tasks like ours, had most likely to do with the very nature of these models and the way they learn. More precisely, a typical ML or DL approach might be rather capable on simultaneously incorporating different aspects of the language, used in a given text passage (lexical, syntactical, semantical etc.), and on identifying all those underlying language patters that govern a given text. In that sense, such approaches might be excellent candidates in a scenario where the model, as an example, should identify words that are important for the overall comprehension of a text passage (that is the example of a typical comprehension exercise type, namely *cloze*, where a text passage is demonstrated to the learner with comprehension-important words having been blanked, and the learner must identify those words based on the comprehension of the overall context of it). On the other hand, a grammar generation task might be considered as a much more straightforward one, since in that case, simple and direct syntactical or morphological queries could sufficiently isolate a target grammatical construction from a given corpus and in a similar manner, convert it with the incorporation of handcrafted transformational patterns. As a result, we considered a rule-based approach as rather suitable for the task in hand, the implementation of which, will be thoroughly presented on the next section of this chapter.

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.

In the previous chapters, 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 could 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 resources in use. In that way, the Generation module of SemiGramEx can perform a set of queries on the input resources, to retrieve sets of relevant sentences for a target teaching goal and a target difficulty level. Having acquired such sentences, those

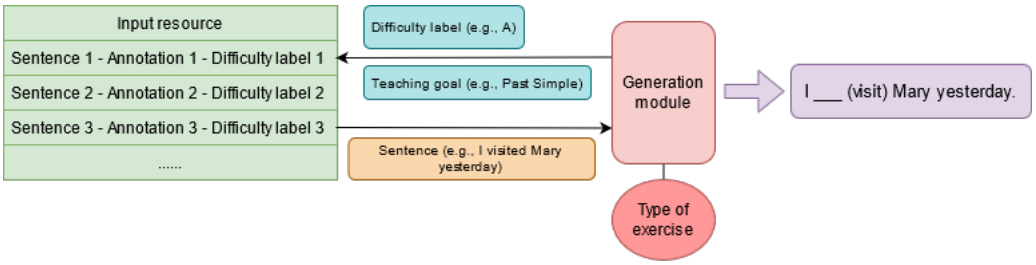


FIGURE 4.1: A detailed exercise generation process of the Generation module.

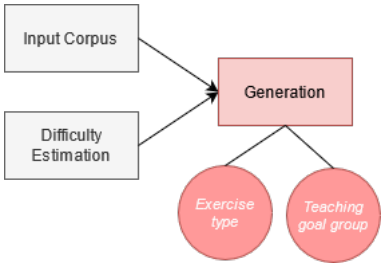


FIGURE 4.2: The Generation module of SemiGramEx.

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, a pivot table of all the subsequently reported details is also presented, to facilitate the overview of all the existing generation parameters.

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* type (FIB), the *Multiple-choice* type and the *Find-the-mistake* type of exercise. In addition, it exclusively targets *Verb Tenses learning* and the existing 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 groups 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 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, 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

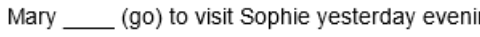


FIGURE 4.3: Example of a FIB type of exercise for Past Simple tense.

the genres or topics that are mostly leveraged in our input resources. 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 exercises that target the simultaneous teaching of each one of the verb tenses existing 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, with the default option being to demonstrate the generated exercise instances in ordered exercise sets. To understand more thoroughly the above, we might consider a generation scenario with *FIB* being the exercise type, *Present Simple and Present Progressive* as the teaching goal group and *ten exercise instances* 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 the verb tenses separately, or they could be demonstrated as a shuffled set of exercise instances, targeting the teaching of both tenses in a mutual setting.

Eventually, concerning the supported types of exercise, the first one that was incorporated in our framework was that of FIB exercises. The FIB type of exercise is perhaps one of the most frequently used exercise type that a L2 learner might have to confront in a L2 grammar learning setting. It is built with the removal of a word from a target sentence 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 tenses 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 sets of the target input resource so that only sentences containing a verb in a target verb form, are retrieved. That iteration continues until a subset of sentences corresponding to a target number of exercise instances and a target teaching goal, 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 that teaching goal group. Having acquired a set of such sentences, SemiGramEx identifies the index position of the target verb in each of those sentence (as well as the depended auxiliaries if any) and converts that target verb into a FIB exercise. More specifically, the target verb along with any possible auxiliary are deleted and being replaced by a blank, while also the lemma form of that replaced target verb is demonstrated in a

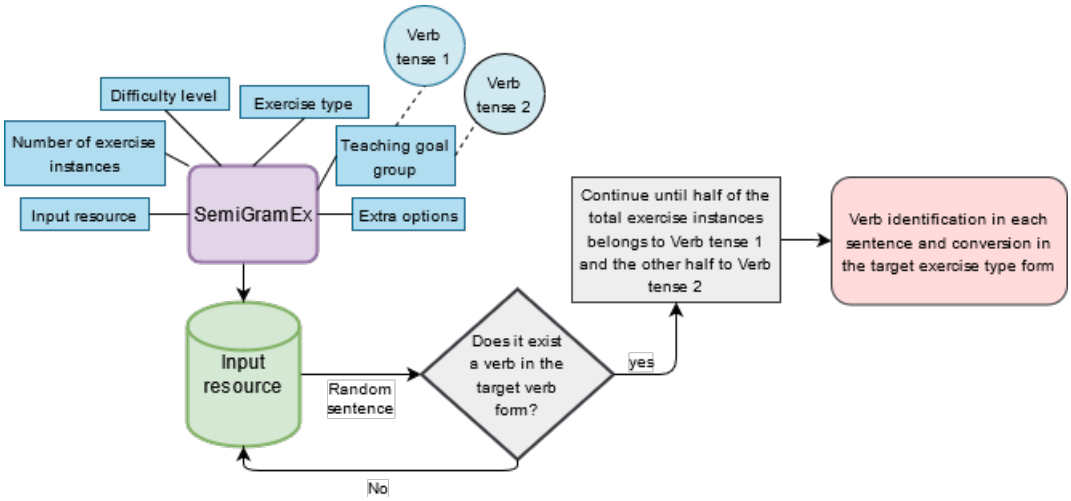


FIGURE 4.4: The detailed process of retrieving sentences before converting them into exercises.

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

FIGURE 4.5: Example of a Multiple-choice type of exercise for Past Simple tense.

parenthesis to the learner, as a hint. All the aforementioned 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 constructed in a similar manner with the FIB exercises, 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 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 words that are being replaced with a blank and the target of our interest, are the verbs and the aim of such an exercise type, is to practice and evaluate *verb tenses identification*, where the learner should be able to 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 verb along with any possible depended auxiliary 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. In fact, the distractor that is demonstrated by SemiGramEx is the verb of the other verb tense that

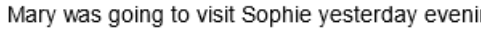


FIGURE 4.6: Example of a Find-the-mistake type of exercise for Past Simple tense.

exist on the target teaching goal group. To clarify that, we might consider as an example the generation scenario of a *Multiple-choice* exercise type and *Present Simple and Present Continuous* as a target teaching goal group. In that case, SemiGramEx will initially retrieve a subset of sentences containing a verb in the present simple verb form. That verb will then be replaced with a blank and demonstrated as an option. The same verb will then 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 with the conversion of well-formed sentences into ill-formed ones, by injecting mistakes on them. Having such ill-formed sentences, those can be demonstrated subsequently to the learner, in a mixed collection of ill-formed and well-formed sentences. Many possible variations of such an exercise type might exist, where the learner might just have to indicate whether a sentence is true or false, or even specifically specify the existing mistake and correct it. 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, that 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 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 a generation scenario where a *Find-the-mistake* exercise type is selected, along with a number of *ten exercise instances* and *Present Simple and Present Progressive* is the target teaching goal group. In such a case, half of the 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 a final exercise instance.

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 those two verb tenses (i.e., the present perfect and past perfect) are not easily distinctive 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).

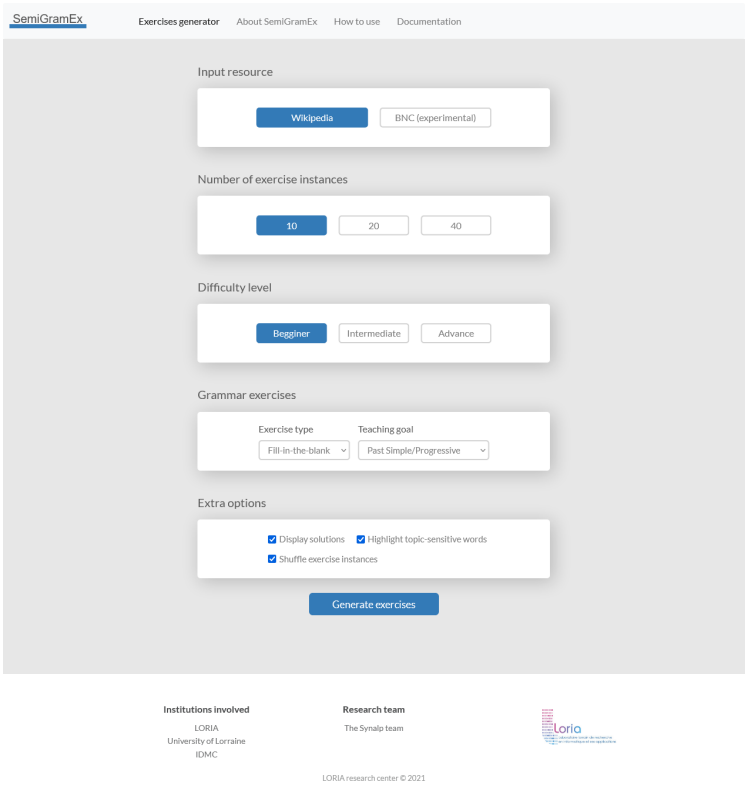


FIGURE 4.7: The SemiGramEx web-interface.

4.2.2 Overall presentation of the SemiGramEx UI

Until 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 those components. The reason for that, is to present a 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 in the future. The SemiGramEx interface is demonstrated on Figure 4.7. In addition, a table of the main supporting parameters is demonstrated on Appendix B.

All the necessary information that surrounds 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 nature of this project, tips about the meaning of all the interface 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.

In addition, concerning each one of the generation components existing on the web-interface:

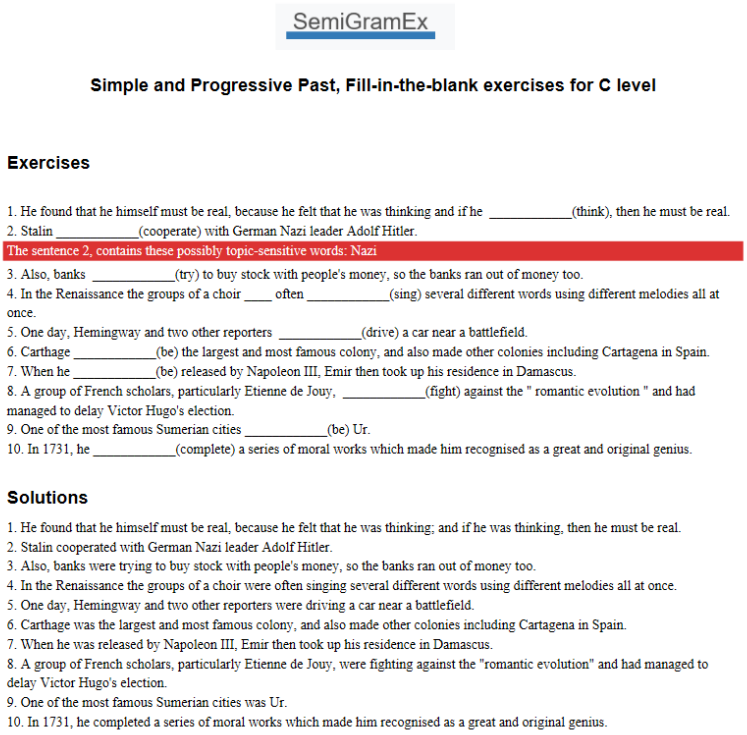
- **Input resource**: That parameter concerns the input resources that are incorporated as a generation basis of SemiGramEx and were reported on Chapter 2. The Wikipedia option employs the Simple Wikipedia as a generation basis, while the BNC option employs the small British National Corpus resource as a generation basis. It should be noted that the BNC corpus results are still not optimal, and this option is only provided for a plurality of 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 ten, twenty and forty 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 proficiency level that the final generated exercises, should maintain. The process of creating such a difficulty estimation component, was reported on Chapter 3. Three difficulty levels are supported for the time being, those being the Beginner, the Intermediate and the Advance proficiency level and the aim of this module is to aid the teacher, by classifying the generated results on those three difficulty levels. However, it should be mentioned that during the implementation of this module, we had strongly assumed that a L2 teacher must interfere and improve any possible deficiency of our difficulty estimator, meaning that the results of this difficulty 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 generated exercises, 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 part, were previously reported on this Chapter.

- **Extra options**: Those parameters aim to further aid the teacher to acquire a customized and personalized user experience. More specifically, the selection of the *Display solutions* option, will demonstrate the solutions of the generated exercises, along with the generated results. The *Highlight topic-sensitive words* option will display an indication of possibly inappropriate words that exist in the generated exercises, if any (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). Eventually, the *Shuffle exercise instances* option, shuffles in a random order the final generated results, 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



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 4.8: A set of FIB exercises generated by SemiGramEx in a pdf format.

tense. 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 an equal number of generated exercises both for the present simple and the present progressive verb tense, 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 form, the Shuffle exercise instances option can be remained unselected and, 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. In that way the teacher can isolate the wanted results.

Conclusively, having selected all the aforementioned options, a set of L2 grammar exercises will be generated, according to the chosen parameters. To display the generated results, we have created a dedicated component that is responsible of creating a well-structured *pdf file*, where the generated results should be demonstrated. An example of such a file might be seen on Figure 4.8. The chosen parameters for that example, were the *Wikipedia* as an input resource, a number of *10* exercise instances, a *Beginner* 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 indicated that the generation procedure might take some moments, depending on the selected parameters, since for the moment it can only be executed in a local machine. In the future, the SemiGramEx project might be hosted online in a dedicated server with adequate processing power to instantly demonstrate the generation 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 final outcomes of 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. Since the main target of our work is the human, what interest us the most in this evaluation part is the human-based assessment while the automatic evaluation part was mostly conducted for a plurality of insights.

However, 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 period, had unfortunately rendered unfeasible an extensive human-based evaluation approach. Given such constraints, we were only able to employ 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 taken carefully under consideration. In an ideal scenario, besides an adequate number of evaluation participants we would also opt to evaluate our framework on 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 the most time-consuming tasks. Moreover, in a similar manner, the suitability of our framework could also be tested on the performance and 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.

5.1 Automatic evaluation on the Difficulty Estimation module

Starting with the first part of our evaluation, we have conducted an automatic evaluation for the Difficulty Estimation module. As we have already explained on Chapter 3, we treated the difficulty classification problem in hand, as a multi-class classification task. More precisely, a biLSTM multi-class classifier was trained on a custom

	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 classification metrics values for the biLSTM difficulty classifier.

training corpus, so that a sentence could be given as an input and a difficulty class could be assigned as an output. As a result, some of the most well-known evaluation metrics for such classification tasks can be incorporated to automatically evaluate the results, with those being the *Accuracy*, the *Precision*, the *Recall*, and the *F1-score*.

Briefly on those metrics:

- The Accuracy metric measures the rate of correct classifications.
- The Precision metric measures the proportion of instances (in our case sentences) that turns out to be correct, in the group of instances that are 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 on 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 relatively similar values with the Accuracy score, 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 in Figure 5.1, the training and validation loss of our model was naturally decreasing over the 7 training epochs, with the final training loss of the model being around 22.2 and the final validation loss around 22.6. Those two values were adequately low and in a balanced correspondence with each other, indicating that the model did not overfit on any of the training or the 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 further evaluate the results of our Difficulty Estimation classifier with some external, and perhaps more objective, measures. However, as it might have already been understood by the literature overview that was reported on Chapter 3, it would be quite difficult to compare our classification results with an existing classification approach. To our knowledge, most of them did not concern L2 readability assessment on a sentence-level, and those who did, had incorporated ML techniques and/or a different difficulty scale than ours. Therefore, we decided to further evaluate the generalization capability of our model, with the leverage of two sentence-level L2 readability assessment corpora. The first one of those corpora was the *OneStopEnglish* [39] corpus which was previously presented, and the second one was the *Sentence Corpus of Remedial English* (ScoRE) [44], 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.

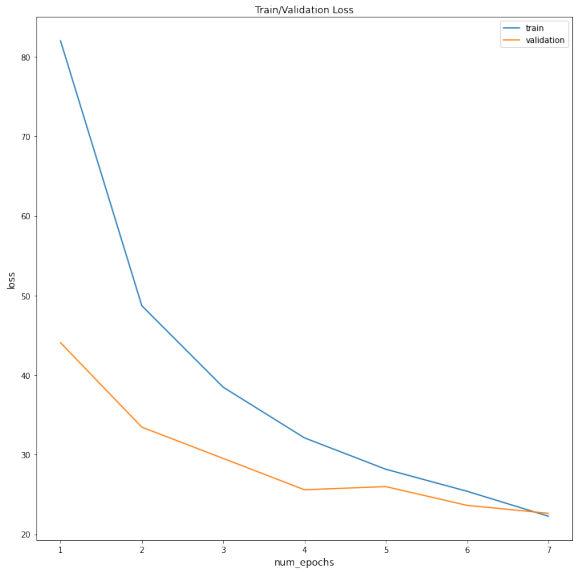


FIGURE 5.1: The training and validation loss of our difficulty classifier.

	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 values of the classification metrics on 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 an 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 of the model' s learning process had managed to acquire high Accuracy results, the generalization experiments on those two corpora did not point on the same direction. More specifically, a 37% Accuracy score was obtained for the OneStopEnglish corpus and a 43% for the ScoRE corpus. The F1-score results have also demonstrated a great variability, with the macro average score being 33% for the OneStopEnglish corpus and 40% for

	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 values of the classification metrics on the ScoRE corpus.

the ScoRE corpus. Those results are rather poor, and in any case much lower than what we expected, or in comparison to what was presented at Chapter 3 as the usual literature results for similar tasks.

However, even though the Accuracy and the F1-score measures might be the desirable classification metrics in general terms, 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 task is in any case quite vague, 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 OneStopEnglish corpus' results for the Beginner level were also better than the next best difficulty level (Advance) in a percentage of 22%, even though its results were much lower than those reported on the ScoRE corpus.

For a better understanding of these results, we conducted a more thorough examination on the difficulty annotations existing on those two evaluation corpora and the wrong classifications that were made by our framework. However, to facilitate ourselves, we 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.

Concerning the OneStopEnglish corpus, the first thing that was observed, was that it contained many cases where sentences with an extensive length 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 a small part of the sentence, 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 extend 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: Beginner sentence annotations examples and their model predictions, for the OneStopEnglish corpus.

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: Difficulty sentence annotations examples and their model predictions from the OneStopEnglish corpus.

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

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

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: A sentence tuple annotation example of the difficulty hierarchy from the OneStopEnglish corpus.

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: A sentence tuple annotation example of the difficulty hierarchy from the OneStopEnglish corpus.

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. 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 in that corpus. 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 the sentences of Advance level, since for such a level the model might primarily expect a sentence of a greater length. On the contrary, the results on the OneStopEnglish were poor in all cases, due to the deficiencies that were previously mentioned.

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 quality of our framework. More precisely, what is mainly being proved with all these diverse automatic

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 advance sentence annotations examples from the ScoRE corpus.

results in our opinion, is that such objective measures might not be the most suitable way to evaluate the elusive 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 should trust the most. Between the high result scores obtained from the training procedure and the lower result scores obtained from the two aforementioned 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 on 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 the teachers were asked to estimate which difficulty level, from the pre-existing three 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 similar manner, as what they would have done during their duties in a L2 educational setting (meaning that no strict guidelines were given, but rather the intuitive method they would follow for such a task was demanded). The purpose of that evaluation section was to study the correspondence between the difficulty estimations made by a L2 teacher and the estimations made by our model, for a given set of sentences. All the sentences that were demonstrated in this evaluation part, were randomly retrieved from our 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 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

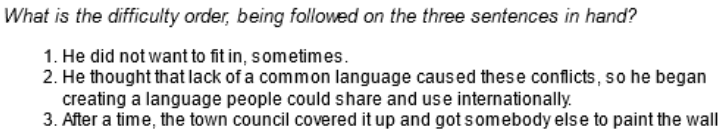


FIGURE 5.2: A triple of a difficulty hierarchy evaluation example. A possible answer might be 1:A, 2:C, 3:B.

see whether the model was able on properly distinguish three sentences on the corresponding difficulty categories, 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 have done in a real-world teaching setting. All the sentences that were demonstrated in this part, were also randomly retrieved from our corpus.

5.2.2 Evaluation on the Generation module

Besides assessing the quality of the Difficulty Estimation module, we have also conducted a human-based evaluation for the final exercises' results, produced by the Generation module. For that part of the evaluation, the evaluators were the same 4 teachers as with the evaluation part of the Difficulty Estimation module, separated in two groups 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 for each of the two evaluator groups. Moreover, the difficulty level and the target teaching goal for those exercise instances were chosen randomly since time limitations had restricted us on intensively evaluate all of them. Eventually, all those exercise instances were retrieved from the Wikipedia input resource (we ignored for the moment 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, the *ambiguity* of our exercises (i.e., the degree of context-dependence and the precision in terms of meaning), the *appropriateness* in terms of a target teaching goal, and the *preparedness* of 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 are 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 would be revealed. Having conducted all this, we were able to evaluate 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.

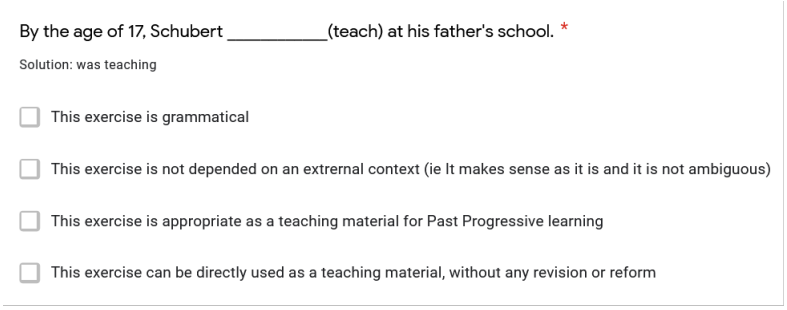


FIGURE 5.3: The 4 possible evaluation options, that were demonstrated for each of the exercise instances.

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.

5.2.3 Results on 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. 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 a human estimation. Therefore, the first insights on these results indicated that, while the framework's difficulty predictions had some distance with the human predictions, 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.

To further stretch the credibility of the above statement, 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 coil 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 1 and the value 0 is equivalent to chance agreement. A score that is less than 0 indicates that there is no agreement. 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 [45], 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 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 of acceptability. 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 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 reported percentage of this dimension is utterly encouraging, indicating that the framework is already in a much better status than what we had initially thought of.

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 learning.	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.

The only low result in this evaluation section is 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* module we had incorporated (thoroughly presented on Subsection 2.2.3), was not sufficient to 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 have expected that most of the sentences would have 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 exercises were initially context-depended and ambiguous in the way they were demonstrated during the evaluation, it would 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 the 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 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 point on 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 solely the difficulty level of a 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. On the other hand, the decision of a text passage's difficulty level, in relation with another text passage, might be a much more straightforward task.

4. Most of the teachers reported that sentence length was a decisive factor on the difficulty estimation process, both for the first and the second evaluation section. That comment is in correspondence with the behavior we have previously reported on our framework, concerning the sentence length.

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 an 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, would render them suitable for a L2 teaching setting.

6. All the evaluators had estimated that SemiGramEx was generally rather useful as a support on their duties, and all of them had reported an excellent 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 L2 grammar exercises for verb tenses learning. A collection of English sentences was leveraged as a generation basis for that purpose, which was 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 and retrieve sentences based on a specific target teaching goal, before converting them into appropriate grammar exercises. Rule-based techniques and handcrafted patterns were incorporated for that purpose and three types of exercises are supported, with those being the Fill-in-the-blank, the Multiple-choice, and the Find-the-mistake type of exercise. In addition, a difficulty estimation 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 that we created, that accepts an English sentence as an input and results a predetermined difficulty class as an output. Eventually, a set of extra parameters 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 teachers. More specifically, we have separately evaluated the difficulty estimation component and the results of the final generated exercises. For the former, the results we have obtained from various automatic evaluation metrics 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 subjective and elusive, even for a human. More specifically, it appeared that while the difficulty estimations performed by the framework were only in a medium correspondence with the estimations that were performed by the humans, an extensive disagreement was also demonstrated between the evaluators themselves, suggesting in that way the subjective nature of the 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 in our estimation, that should be ameliorated in the future.

Starting with the minor improvements:

- Different input resources could be incorporated as a generation basis to extent the supported teaching goals, since for the moment, only 6 verb tenses are supported, which are those that mainly exist in the input resources in use.
- An extra input resource option could be employed, where the teacher could use an input text of preference as a generation basis. For this to happen, an initial issue that should be overpassed is to decide about the accepted text form

that would be accepted in such a case (i.e., a limited input text field, or a complete pdf file, etc.). Furthermore, another issue might be to retrieve ways that such an input text could be preprocessed (since the pre-processing techniques we have incorporated were developed in a "trial and error" spirit, according to a target text each time, rather than techniques that could be applied with valuable results on generic-purpose text passages). Eventually, another major issue might be to find ways that such an input text could be parsed with the relevant linguistic information (in this work, the parsing techniques we have leveraged were executed on a previous step of the generation procedure, since they would require an extensive amount of parsing time to be performed "live" before every generation).

- 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. If a more extensive lexicon could be retrieved, it might 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. With that option, the generated results can be demonstrated in a mixed group or in two ordered groups, where each group corresponds to each teaching goal. 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 input sentences in use. That deficiency mainly derives from the way that our input resources have been leveraged, since those input resources 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 maintain with their surrounding context. To resolve that issue, a sentence selection module was employed, which was extensively presented on Subsection 2.2.3. However, the filtering conditions we had applied at 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. Furthermore, that context-dependence deficiency has also demonstrated a great effect on specific types of employed exercises. More specifically, 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 exercises might still suffer in a great extent from 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, both the solution options might appear to be suitable). An example of such an issue for a Find-the-mistake exercise, can be seen in Figure 6.1. To

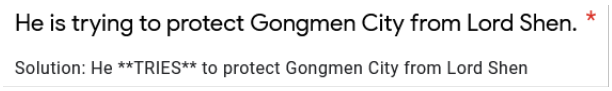


FIGURE 6.1: An ambiguous Find-the-mistake exercise, where both of the incorrect (Present Progressive) and the correct (Present Simple) tenses, could be correct, depending on the surrounding context.

resolve that issue, perhaps a more extensive incorporation of the filtering choices, being reported on [15], 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 be evolved into a language-independent generation framework, or at least a multilingual one, in case that the language-independent goal is ultra-optimistic. While the initial objective of this work was to implement a generation framework that would be able to also function on other languages besides English, the nature of a grammar exercise generation task appeared to be much more restrictive than what was initially thought of. More specifically, we have searched for ways that the rule-based generation techniques in this work, could 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. However, there is always a 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 further 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 ones that were incorporated in this work. In addition, another extension of this project would be to also acquire a user-side that would specifically concern 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 grade 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 grade lexicon.

Appendix B

SemiGramEx main supporting parameters

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

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

Bibliography

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

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

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

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

[5] Ayako Hoshino and Hiroshi Nakagawa. "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*. 2005, pp. 17–20.

[6] Alena Fenogenova and Elizaveta Kuzmenko. "Automatic generation of lexical exercises". In: *Proceedings of the International Conference*. 2016.

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

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

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

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

[11] Michael Heilman and Noah A Smith. "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*. 2010, pp. 609–617.

[12] Antoine Chalvin, Egle Eensoo, and François Stuck. "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*. 2013, pp. 280–295.

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

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

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

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

[17] Council of Europe. Council for Cultural Co-operation. Education Committee. Modern Languages Division. *Common European Framework of Reference for Languages: learning, teaching, assessment*. Cambridge University Press, 2001.

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

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

[20] Luo Si and Jamie Callan. "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, 2001, pp. 574–576. ISBN: 1581134363. DOI: 10.1145/502585.502695. URL: https://doi.org/10.1145/502585.502695.

[21] Kevyn Collins-Thompson and James P Callan. "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*. 2004, pp. 193–200.

[22] Sarah E Schwarm and Mari Ostendorf. "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)*. 2005, pp. 523–530.

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

[24] Michael Heilman et al. "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*. 2007, pp. 460–467.

[25] Sowmya Vajjala and Detmar Meurers. "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*. 2012, pp. 163–173.

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

[27] Wade Shen et al. *A language-independent approach to automatic text difficulty assessment for second-language learners*. Tech. rep. Massachusetts Inst of tech Lexington Lincoln lab, 2013.

[28] Thomas François. "An analysis of a french as a foreign language corpus for readability as-sessment". In: *Proceedings of the third workshop on NLP for computer-assisted lan-guage learning*. 2014, pp. 13–32.

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

[30] Luciana Forti et al. "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*. 2019, pp. 360–368.

[31] Ildikó Pilán, Elena Volodina, and Richard Johansson. "Rule-based and machine learning approaches for second language sentence-level readability". In: *Proceedings of the ninth workshop on innovative use of NLP for build-ing educational applications*. 2014, pp. 174–184.

[32] Sowmya Vajjala and Detmar Meurers. "Assessing the relative reading level of sentence pairs for text simplifi-cation". In: *Proceedings of the 14th Conference of the European Chapter of the As-sociation for Computational Linguistics*. 2014, pp. 288–297.

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

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

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

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

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

[38] William Hwang et al. "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*. 2015, pp. 211–217.

[39] Sowmya Vajjala and Ivana Lučić. "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*. 2018, pp. 297–304.

[40] Luise Dürlich and Thomas François. "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)*. 2018.

[41] Aina Garı Soler, Marianna Apidianaki, and Alexandre Allauzen. "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*. 2018, pp. 499–508.

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

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

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

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