Analysing and enhancing text comprehension in the Italian language

Make texts more accessible to everyone with the help of technological tools

Jacopo Caratti

Abstract

The primary purpose of this project is to explore and better understand which are the main techniques that allow improving the accessibility of a written text. These techniques can be applied to any text, or - as in this case - to a text in Italian, and provide a simplified version. Then there is the analysis of some existing solutions, both automated and manual, with some examples of use. Of these, its strengths, weaknesses, and fields of use are described. Finally, the developed technological solution is presented; it uses some of the techniques analyzed to improve the accessibility of a text in Italian. This project aims to help all those people experiencing difficulties to make sense of a text for a variety of temporary or permanent reasons (e.g., intellectual disabilities, elderly, foreigners and immigrants...). This does not limit the understanding of people who do not have these types of difficulties.

Advisor
Prof. Dr. Monica Landoni
Assistant
M.Sc. Leandro Soares Guedes

Advisor's approval (Prof. Dr. Monica Landoni):   Date:

Contents

1 Introduction

Reading and understanding a text is a complex process, with which we are confronted daily in many areas of social life. This process begins with by seeing the character. Here, our ability is to being able to distinguish and identify it, making it possible to group multiple characters in order to generate a meaningful sentence. In the scientific field, we refer to these operations with three specific terms: accessibility, recognisability, and interpretability.

To whom it is addressed

Individuals are all different, each with their own strengths, weaknesses, characteristics and abilities. Whole books and essays have been written on that subject and many others will still be written, but this is beyond the scope of this paper. Suffice it to say that some of us suffer from problems relating to the understanding of written texts. I'm not just talking about books, but - as we'll see - about websites, applications, posts in social networks, brochures, advertisements, etc.

There are many categories of people affected by this type of problem. There are **the elderly**, who very frequently suffer from vision problems, and therefore need a simplified presentation of the text in order to grasp the content. Senile dementia is another problem that often afflicts them: so it could also be important to extrapolate the essential information from a text, so as not to cause more confusion.

There are **foreigners**: people who do not know - or know very little - our language. Especially in the last decade, here in Europe we have closely witnessed immigration from countries across Africa. People fleeing war and an unworthy lifestyle, looking for a new place to live. Often they land on the shores of our continent, but they remain disoriented because, unable to communicate, they don't have any opportunity to make their way into society. There are many categories of foreigners, but the problem they have in common is the same. Therefore, facilitating document texts and developing new solutions for learning the written language is of fundamental importance: to acclimate faster to the new environment and to show their value in the society without impediments.

Then, some of the population suffer from **intellectual disabilities (ID)**. The *American psychiatric association* [5] asserts that these people manifest "*problems with general mental abilities that affect intellectual functioning area (such as learning, problem-solving, judgement), and adaptive functioning area (activities of daily life such as communication and independent living)*".

The level of intellectual abilities can be measured in different ways: standardised tests became very popular in the last century, the most famous of which is certainly the IQ score. The average person's IQ score turns out to be around 100[1], while that of a person with ID is associated with an IQ between 70-75 [5]. Regrettably, for many years there has been the prejudice that individuals with low relative IQs could not learn to read [12]. Fortunately, this misconception is only linked to the past. In fact, it has been corrected in recent decades, during which it has been realised that even people with ID and relatively low IQ could learn to read.

The importance of accessibility

Nowadays, given the omnipresence of written contents, the ability for an individual to access them is of fundamental importance for full participation in social life. This aspect was in fact reported by the UN in the 2006 *Convention on the Rights of Persons with Disabilities* [3], whose *Article 1* ends with:

Art. 1 [...] Persons with disabilities [...] which in interaction with various barriers may hinder their full and effective participation in society on an equal basis with others.

The only way to avoid this is to make the text more accessible to everyone, for example, by simplifying it. It is important to underline that this process of elaborating a text is not at the expense of people without particular difficulties: what we could understand from reading the text, we can safely continue to do so. Furthermore, simplified

---

[1]https://www.verywellmind.com/what-is-the-average-iq-2795284

text is much easier and faster for everyone to scan and understand.

Where it can be useful

The act of simplifying a text is useful wherever there is a text to be read, which, nowadays, we know is everywhere. Just think about how much and what each of us delegates directly to written communication every day: reading and writing messages, reading and writing emails, reading and writing posts in social networks, reading and writing documents, etc. And this only from us to the outside. Because the stimuli of a written nature that we receive daily are also infinite. In addition to those already mentioned there are: reading books, reading articles, reading webpages, understanding important documents - often written in a formal and refined Italian -, being able to fully appreciate a visit to a cultural center, choosing a meal from the menu of a restaurant.

Therefore, it is clear that written language is present in every area of society, moment of our lives. From the simplest category, such as the billboard placed on the way home, to the most highly placed spheres, such as formal documents and articles drawn up by the *Council of State of the Republic and the Canton of Ticino*[2], which individually concern all of us citizens of Italian-speaking Switzerland. Without the simplification of the texts, all this would be out of reach - or hardly within reach - of those categories of people mentioned above, and, often, discriminated against, precisely because of difficulties like this of reading, which marginalise them from living fully in society.

Goals and motivations

The purpose of the project is to develop a technological solution that improves accessibility by providing a systematic lexical simplification of the text in Italian. The project aims to help all the different people that I presented above making access to reading much easier.

A few months ago, I had to choose between different Bachelor thesis proposals. When Professor Monica Landoni presented this project to me, I immediately accepted it. In the first place, what convinced me was the simplicity of the work, but in a good way: doing something useful for people who need help, for them to be able to become part of the - normal for many of us - everyday life. For example, people with ID, yes, they are a minority, but they are still 1% of the world population and they need people capable of developing new methodologies that can prove useful. The project is within my reach, and I am honoured to be able to do my part in this direction.

On an IT level, I really like the idea of creating something that is directly usable by people, something visible and tangible like an application. It has often happened to me to program, but often much of the work goes unnoticed by most, because it is used only in the background. In this case, I think my final work may appear prominently and understood by people outside the IT industry as well.

---

[2]Wikipedia: The government of the Canton Ticino

2 Simplification of texts

I found the most precise and concise definition of *text simplification* on Wikipedia [34]: *'[...] it is an operation used in natural language processing to modify, enhance, classify or otherwise process text in such a way that the grammar and structure of the prose is greatly simplified, while the underlying meaning and information remains the same'.*

Given the complexity of the language, creating fully automatic simplification tools is very complicated: the vocabulary is very wide and sentences could be structured in millions of different ways. What is most often done is to improve the presentation - and therefore structure - of the text, in order to help the readers. In this sense, we can use the so-called *Augmentative and Alternative Communication* (AAC). AAC is about all the communication devices, systems, strategies, and tools that replace or support natural language speech or writing for those who have difficulty producing or understanding them.[3] These techniques are well known, and are also used and investigated by many researchers. In the scientific field, AACs are cited as an excellent approach to making information sources more readable.

Related to AAC, it should be noted that there are two main categories of communication: aided and unaided. With **unaided communication** we mean everything related to the natural communication between individuals - voice, gestures, facial expressions, and also the sign language -, then everything that doesn't need an external support to be used. Differently, we define **aided communication** everything that requires an external support to work as communication medium. Examples are objects, images, pictures, drawings, symbols, and also hardware devices such as computers and tablets, and software such as speech synthesisers [11].[4]

In some papers I have analysed, these techniques are proposed as solution in the cultural field to facilitate access and understanding of information during the visit to a museum [11]. Therefore, I thought of personally going to visit a museum, so as to probe if the information on display could be easily accessible to every type of person. On March 12, 2021, I went with professor Monica Landoni and doctoral assistant Leandro Soares Guedes to the *Museo d'arte della Svizzera italiana* (*MASI*). Expectations on the accessibility of information on the works exhibited were high, in fact the *MASI* is a museum of international standing, located in the center of Lugano. On that day there were three exhibitions on display: *La collezione, Marta Margnetti*, and *Vincenzo Vicari*, spread over three different floors.

At the entrance we were provided with brochures regarding the installations. You can see part of it in Figure 1. The style and format used in them made it difficult to read: too much information, too long sentences, minimum spacing between lines, different fonts, of different sizes and colours, many symbols.
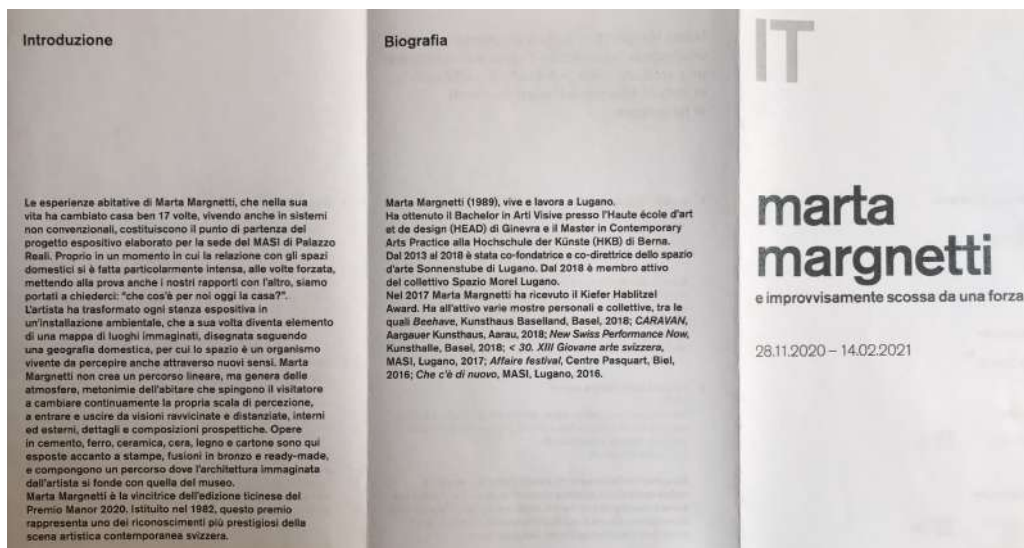

Figure 1. Part of the Italian brochure related to the *Marta Margnetti* exhibition.

This problem was not solely related to the brochures. In fact, every text in the museum had the same style: elegant but complex. I made a collage of some of the pictures I took of the different texts found at the museum. You can find it in Figure 2. In one case, it has also happened to find a work from the exhibition, but not being able to

---

[3]https://www.assistiveware.com/learn-aac/what-is-aac
[4]https://www.asha.org/njc/aac/

identify what it was because of the absence of the captions. After some research, the caption was then identified in a position away from the work, where, as a visitor, I would never have thought.

Figure 2. Some of the texts present in the three exhibitions at *MASI*.

This visit allowed me to start thinking about how to make a text more accessible to everyone, and which features and elements should be avoided or kept in order to make it easier to read. For example, the fact that some dates were extrapolated and underlined could be a good idea: this way the reader immediately know which period the text is referring to. This technique was also used to highlight names of authors and titles of the paintings on the exhibit.

I consider it important to emphasise that the text simplification techniques I will present are inherent to the digital dimension. Therefore, I have left out all those strategies that apply rather to non-digital texts: for example, to the use of A4 format sheets. It is proven that the use of this type of format facilitates the understanding of a written text [11], however, in a digital support such as the one I will later develop, it makes no sense to talk about sheets format. But of other types of expedient, then I will limit myself to presenting these latter.

In addition, all these techniques come from scientific studies on the subject of learning [12, 11, 19]. These studies were carried out according to the modern approach of **inclusive research**, in which, in addition to experts in the field, people directly affected by the problem of written communication expressed themselves on the matter, presenting what for them could be the best strategy for simplifying the text. To complete that, it is important to underline that the meaning of the text should not be changed, otherwise the simplification has not happened in the correct way. This is more difficult in the case of automatic tools, as they very often require no human supervision.

2.1 Manual techniques

In paper [12] there is an example of a real simplification of a text made after the visit to the *Natural Science Museum*[5] in Trieste. Here, I decided to present to you the original text, a first version simplified by professional translators, and two additional simplifications carried out by different groups of people following the inclusive research pattern. Red terms are considered difficult to be read, so they are subsequently replaced with green terms, which in some cases are added without substituting any word but only to make everything more accessible. Finally, the symbol "↓" means a new line has been added.

---

[5]https://www.turismofvg.it/museums/museum-of-natural-history-of-trieste

> **Original version**
> Do bats only live in caves?
> No! Many species of bats routinely take refuge in the cavity of trees or in cracks between rocks. It is precisely these species that often enter households.
>
> **First simplification - Carried out by professionals linguists**
> Bats live in caves and tree holes. ↓
> Some types of bats have become accustomed to live ↓
> also near houses and cities.
>
> **Medium simplification - Carried out by direct users**
> Bats live in places where it's dark. ↓
> For example, in caves.
> Bats can also live near houses,
> and can also live in attics and cellars of houses.
>
> **High simplification - Carried out by direct users**
> Bats live in places where it's dark.
> For example, in caves. ↓
> Caves are underground.
> Bats can also live near houses,
> and can also live in attics and cellars of houses.

From the definition of text simplification presented at the beginning of Section 2, the main aspects considered during simplification are mainly two: **structure** and **grammar**, which are addressed into the following two Subsections. Inside them, when possible, I will refer to this example of already simplified text to better illustrate the processes to you.

Structure

The task delegated to the structure of the text is to present the information in a format that makes it the most clearly readable.

Many reports of scientific studies that I have read on the subject [11, 12] converge on the same following precautions. It has been shown that the choice of spaces greatly influences the readability of the text. Adequate space between the lines is of fundamental importance: the one considered best was **double spacing**, with **margins of 2.5 cm** from the edges of the screen. Just look at the times it was decided to add a new line in the example. This consideration also applies to the spaces between letters in words, which should not be too much, nor too little. In this sense, the choice of the font also plays a fundamental role: it is necessary to use it as simple and clear as possible, avoiding italics and too convoluted characters. The **font Arial 14** was elected the best to read.

It is also important that the text is clearly visible: therefore prefer **homogeneous backgrounds**, and a **font color** that clearly contrasts (the classic white background with black text - or the opposite as in Figure 3 - is always great!).
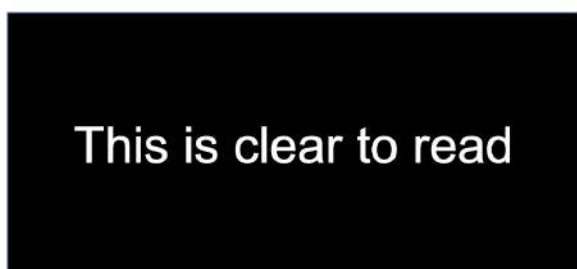

Figure 3. The same sentence *'This is clear to read'* presented in an easily accessible way (left), and not (right).

A good idea is also to provide support for more complex terms. This can be done in the form of explanations, definitions and **synonyms**, which help in understanding the word and its context [19]. In fact, as you can see in the

example, many terms are considered difficult to understand. Some of these have not even been noticed by professional translators.

Finally, one of the most important aspects is the use of **images and/or symbols** that accompany the text. Clearly, these images are not chosen at random, but specially designed to be recognised at a glance: I see the image and immediately know the topic of the text, or, I struggle to grasp the meaning of a word, but the symbol associated with it clarifies my ideas. This text-images association is a metacognitive strategy - just like the **vocal support** that can be provided for each term - to facilitate the understanding and processing of information. Moreover, images are more related to an emotional factor and ease the process of memorisation.

Grammar

The transformation of text into a grammatically simplified version is not an easy operation. It required - and even requires today - empirical research to investigate the processes of readability and comprehensibility in their complexity. Readability is more associated with the presentation of the text, so mainly to the structure, whereas comprehensibility with the content, then how we can simplify the way the text is written to make it more comprehensible [12].

The first thing we want to consider is the appropriated **level of simplification** of the text: do we want it to be universally accessible? Do we want it to be accessible only to those who have certain knowledge, so only a slice of the population? In fact, simplification can take place at several levels, which broadly correspond to the *Common European Framework of Reference for Languages* [26]:

- A1: text very easy to understand

- A2: text easy to understand

- B1: text understandable by average people

Already in the proposed example, two levels of simplification are presented: medium and high. In the original paper [12] these levels are due to different groups of text analysis according to an inclusive research process. The group that later developed high simplification was the one with the greatest learning difficulties.

Numerous studies on the subject agree that people with reading difficulties prefer **reduced texts**, without too much information, which can certainly create confusion. This clearly influences every part of the text, in fact **short sentences and paragraphs** are preferred. But how does this happen? Because reducing a text, we must necessarily eliminate some elements. We proceed by making a selection of the **most important contents** and the most salient aspects. By looking at the example, you may have noticed that the original version of the text contained almost only 2 long sentences, whereas the high simplified one contained 5 short sentences. The content was the same.

Once the level of simplification has been defined, we move on to analyse the lexicon used: too technical terms must be replaced by **high-frequency words**, so that their meaning is clearer. For instance, in the bat text, the not-so-common term *households* has been replaced by the very common *houses*.

Another element to take into account are the **verb tenses** used. Obviously, the simpler ones are preferred. By simpler we mean present simple, past simple, and future simple, and their counterparts in other languages. For instance, in Italian we avoid using the *passato remoto*, in favour of the *passato prossimo*. *Condizionale* and *congiuntivo* can also cause confusion, therefore we usually replace the latter with *indicativo imperfetto* and *indicativo presente* respectively. This is done following the guidelines presented by *Anffas*[6] [13]. For those of us who do not clearly remember - or don't know - the Italian tenses, below I have reported some examples of simplification with respect to them.

---

[6]Associazione Nazionale di Famiglie di Persone con Disabilità Intellettiva e/o Relazionale

Indicativo passato remoto → Indicativo passato prossimo:
Dante Alighieri nacque a Firenze. → Dante Alighieri è nato a Firenze.

Condizionale presente → Indicativo imperfetto:
Biancaneve non vorrebbe quella mela. → Biancaneve non voleva quella mela.

Congiuntivo presente → Indicativo presente:
Per quanto ne sappia, la situazione è incerta. → Per quanto ne sai, la situazione è incerta.

To better present a text, we also tend to **avoid special characters**, limiting ourselves to the use of the letters of the alphabet and classic punctuation. For example, in cultural and historiographical fields it is common to come across Roman numbers. These - as indeed any *special* character not belonging to the alphabet - must be avoided, as they require a greater reading effort.

The way **dates** are written is another simplification that can be implemented very frequently. There are many formats to write a date, just think of English that has at least six.[7] It is therefore necessary that a single format be chosen. The best is the one in full, in which the month is written literally, and the year in full.

For a more in-depth presentation of the simplifications that can be implemented, I suggest you to consult the specific *Anffas* guidelines [13] on easy communication. They are very complete on all the aspects concerning the accessible Italian language, and numerous examples on how to apply simplification techniques.

2.2 Automatic techniques

The techniques that I will briefly present in this section refer to the automatic simplification of a text. Systems such as these receive text as input, and, through a series of processes, return a more accessible version of it.

In the past, when techniques such as machine learning and natural language processing were not yet fully accessible, readability indexes were used. A readability index establishes the ease of reading a text, based on a series of features extrapolated from it. In this way, by calculating the index, it is immediate to understand if the text in question needs some adaptation to make it more accessible [25]. There are numerous indexes, which calculate the readability level in different ways. For the Italian language, the first to be used was the Flesh-Vacca (1986), which was soon after replaced by the much more famous Gulpease index [28], which considers the length of the word and the length of the sentence with respect to the number of letters. Its values range from 0 to 100, where 0 indicates the least readability, and 100 the highest. The Gulpease formula [32] is depicted in Figure 4, where the "#" stands for "number of".

$$89 + \frac{300 * \#sentences - 10 * \#letters}{\#words}$$

Figure 4. The formula used to compute the Italian index of readability Gulpease.

Nowadays, automatic parsing, linguistic techniques, and psycholinguistic information analysis are preferred to the use of indices, which are more anonymous, and return a generalization of the text.
After determining whether a text needs simplification, there are two separate actions to be taken. The first is the automatic lexical simplification, while the second is the syntax.

The **lexicon** is defined as the complex of words and expressions of a language. At this stage, you need to identify which words might be causing a problem, and, after careful context analysis, select the best replacement from a word database. It is clear that selecting the best replacement in a given context requires more sophisticated techniques, such as machine learning and NLP.

The simplification of the **syntax**, which concerns the study of the functions of the sentence structure, is perhaps the most complex part of the process. In fact, the blind application of syntax simplification could damage text cohesion. In the past, it was common to define a series of patterns, which when encountered, could be simplified through

---

[7]https://www.englishclub.com/vocabulary/time-date.htm

a series of rules. But this clearly could only be applied to a few sentences. This process clearly requires the consulting of linguistic experts. To avoid problems, parser analysis and statistical machine learning have been introduced. The use of these systems requires a training data set, therefore a database of texts and their simplified version. Patterns are recursively identified and simplified until they are no longer present in the text [24].

In addition to these simpler procedures for simplifying the text, in recent year we moved on the use of artificial intelligence. It is more complicated and requires knowledge at several levels: computer science, statistics and linguistics. These techniques often rely on neural networks, which have shown to perform better than older systems. Older techniques that still be useful and inserted during the process of elaboration of the text, to infer some inherent properties and features. For instance, a text simplification algorithm may apply machine learning to produce some simpler variants of a text that is given as input. These variants may then be processed through NLP tools in order to find out which one is the best: in preserving the meaning and in the level of simplification [21]. These procedures can be carried out in many different ways, but to get an idea of how it is carried out, I present a chronological list of how it can be carried out. This list is extracted from [27] and has been subject to my elaboration.

1. Gather the document to be analyzed

2. Extract from the document the set of elements related with the writing, and the design and layout fields

3. Parse and analyse morphologically and syntactically the set of elements related with the writing field

4. Recognize patterns in each set of elements

5. Infer knowledge using the collection of rules

6. Generate the report about the original document

For a more in-depth reading on these techniques I recommend the book *Automatic Text Simplification* by Horacio Saggion [24]. It navigates through the history of text simplification, providing examples of indices and rudimentary more techniques used in the past. Furthermore, it shows how nowadays through the use application of new techniques we can make texts more accessible at many different semantic levels. Finally, it tells us how to evaluate such automatic systems.

3 Related works

The goal of this section is to present projects that aim to make written texts more accessible to people who need it. The information reported will range from the origins of the projects, what methods they use to implement text simplification, what are their strengths and weaknesses, and in what areas they are used. I will present a Swiss project and a Spanish one, then two of the most known systems at global scale.

3.1 Swiss Project: *Lingua Facile*, Pro Infirmis

- structural and lexical text simplification

- carried out by professionals and ID people

- paid service

The simplification services of written texts were born in the 70s from a cultural movement known as *Easy To Read* [26] and made its way all over the world, up to land here in Switzerland, where today about 800'000 people have difficulties related to reading[8]. The first services in the confederation are offered in the German part (2014), followed by those in the French part (2017), and finally, in 2018, also in the Italian part of Switzerland. In fact, in this year, *Pro Infirmis Ticino e Moesano* inaugurates the *Lingua Facile* (*LF*) service, with the idea that any text can be easier to read and understand if translated into an easy language. *LF* divides the text simplification process into three main parts:

1. Evaluation of the request with definition of the translation level together with the agent
   The agent submits the text to be simplified to *LF*, specifying the desired level of simplification. The request is analysed and if simplification is possible, *LF* proceeds to the next step.

2. Translation
   At this stage the text to be simplified is submitted to some professional translators. They have the task of identifying the most difficult elements, replacing them and in some cases removing them. Furthermore, the text can be rewritten, also depending on the information that is considered most important and therefore you want to keep.

3. Verification by the re-reading group and analysis of the result with the agent
   Here, by following the inclusive research experiment design, the already simplified text is passed to another group of people, which is composed by individuals with reading difficulties. They are included in the analysis process, judging the usability of the product/service from their perspective. The goal is that the final text is suitable for use by all types of people, with and without difficulty. In the case of *LF*, the collaboration with some laboratories of the *Fondazione Diamante* is fundamental. In these laboratories, the texts are reread and commented by ID, so as to remove any residue of too complex Italian, perhaps not considered as such by professional translators.

An example of simplification implemented by *LF* comes from their website, where the article *Le nostre prestazioni principali*[9] has been changed to *Cosa facciamo per aiutarvi*[10]. The entire simplification is reported below, both in Italian and English languages. Pay attention that the translation of both sentences into English is done directly by me, without the consultation of expert English linguists. The aim of this is not to present a simplification of the same text that is also good in English, but to provide a translation to those who are reading and do not understand what is written in Italian.

---

[8]https://www.proinfirmis.ch/it/offerta/ticino/servizio-lingua-facile.html
[9]Le nostre prestazioni principali (IT) = Our main services (EN)
[10]Cosa facciamo per aiutarvi (IT) = How we help you (EN)

> ITALIAN
>
> **Original version:** *Le nostre prestazioni principali*
> Pro Infirmis supporta le persone con disabilità e le loro famiglie con i suoi servizi nella gestione della vita quotidiana e nella partecipazione ad aree importanti della vita, come la vita, il lavoro o il tempo libero.
>
> **Simplified version:** *Cosa facciamo per aiutarvi*
> Vi diamo dei consigli e vi accompagniamo nelle questioni che riguardano: la vita di tutti i giorni, il lavoro, la casa, il tempo libero.
>
> ENGLISH
> **Original version:** *Our main services*
> Pro Infirmis supports people with disabilities and their families with its services in managing daily life and participating in important areas of life, such as life, work or leisure.
>
> **Simplified version:** *How we help you*
> We give you advice and accompany you in matters concerning: everyday life, work, home, free time.

Depending on the desired level of simplification, the techniques implied are different. They range from the use of shorter sentences - as in the example reported - and terms of everyday use, to the juxtaposition of images, but the exact process of simplifying the text is the responsibility of the translators and the laboratories. Here, it is clear how simplification is not an objective matter, but depends on the people involved and the final use of the text, which can have an infinite number of different roles.

The Pro Infirmis website[11], in addition to presenting simplified texts and articles, offers a very useful and interesting service. On any page of the site, by selecting a word or a phrase it will be possible to have it pronounced. This is a further step towards easy understanding of texts.

The services offered by *LF* are certified by the *Lingua Facile* label. This label certifies that the service offered uses and respects the European rules [13] to make the information easy to read and understand. At a European level, another symbol is used to certify the simplification of a text. Both symbols are depicted in Figure 6.
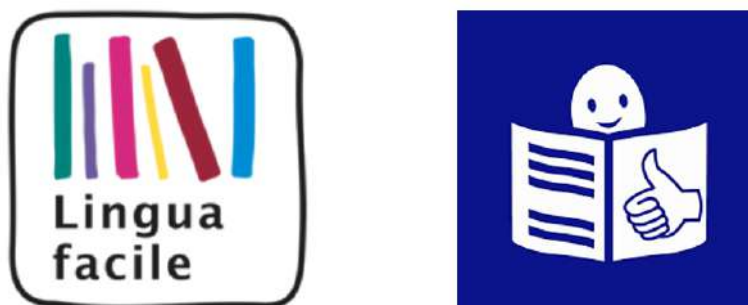

Figure 5. Certification label of *Lingua Facile* (left), and *easytoread* certification label at European level (right).

3.2 Spanish Project: *Easier*, University of Madrid

- support to text

- carried out by an automatic tool

- free service

Easier is a 2019 Spanish project developed by the *Human Language and Accessibility Technologies* (*HULAT*) research group of the *Carlos III University of Madrid*. The project was developed for the simplification of Spanish language texts. However, the techniques used can be extended to other languages, such as Italian, as the two idioms are very similar in terms of syntax and grammar rules.

---

[11]www.proinfirmis.ch

Easier is a web application, but it can also be used as an extension for *Google Chrome* and *Firefox*. It aims to facilitate the accessibility of texts by providing synonyms and definitions for the more complex words. These type of words are identified through the use of artificial intelligence methods, for example - as they mentioned in the project description [16] - *Natural Language Processing* (*NLP*).
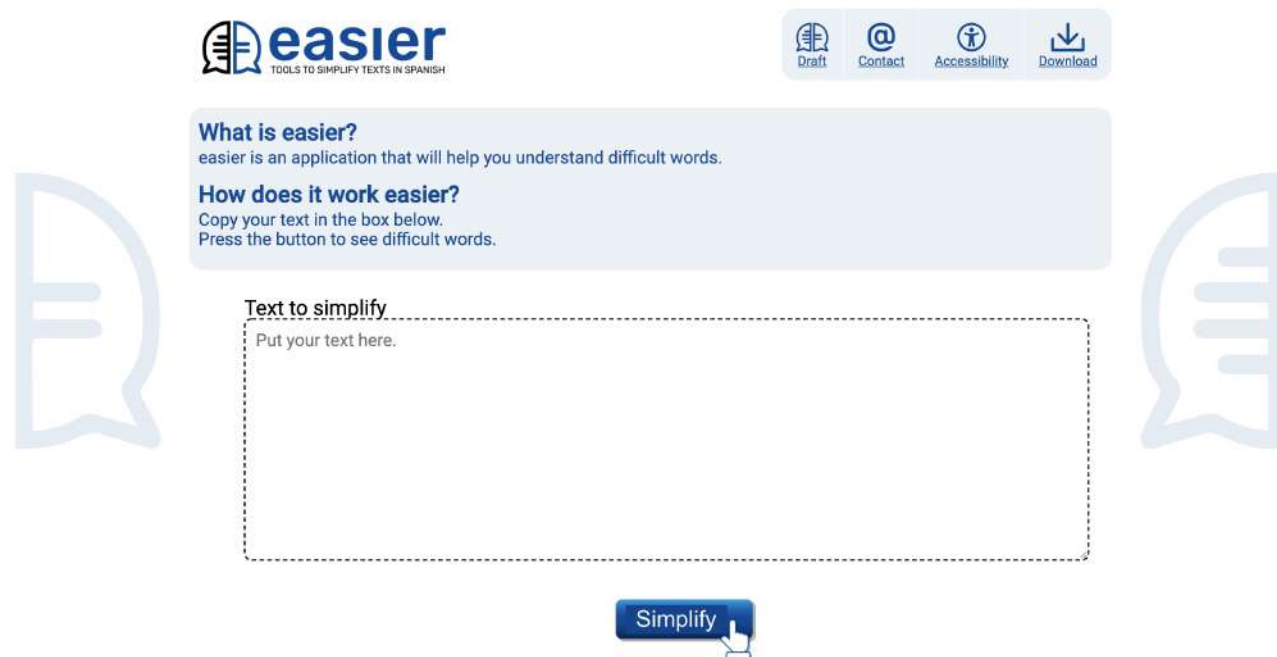

Figure 6. Main page of the *Easier* web application (translated in English).

The web app is in line with the easy-to-read and *WCAG 2.1* techniques: the interface appears intuitive and easy to use. In fact, the elements present in the main page are few and all essential. The first thing that catches the user's eye are the instructions for use, presented in a blue box in the center of the page. Once the user has read them, he exactly knows what to do and how the web application simply works. He copies and pastes his text into the below box and clicks on the simplify button. At this point, the text is analyzed and the words considered complex are underlined in blue, becoming clickable. By clicking on one of these, on the right side they will appear synonyms, definition and - if possible - a pictogram representing the complex term. The operation can be repeated as desired by providing other texts.

I tested the functioning of the *Easier* service by applying the simplification on a piece of Spanish text, randomly taken from a Spanish page[12]. On the left of Figure 7 you can see the processed text, in which the words considered more complex have been underlined in blue. I clicked on the word *dictionary* and its pictogram and definition appeared on the right (complete with the easy-to-read approved symbol).

I continued the test by clicking on the word *volumes*, the result is slightly different and you can see it in Figure 8. The pictogram is absent, some synonyms are present (which were absent in the previous example), and there is a much shorter definition. Comparing the two examples I found that the presentation of complex terms varies according to the term. The only element that is always present is the definition of the word. This behavior is normal, in fact it is natural that there are no pictograms for all the complex words, the same reasoning applies to synonyms.

The way *Easier* project is structured and the techniques that it uses are in line with the easy-to-read standards and *WCAG 2.1* guidelines, making it accessible by all the people.

In addition to the use of artificial intelligence techniques to analyze the text, the system works with databases of synonyms, definitions and pictograms. Synonyms and definitions are extrapolated from the *Diccionarion fácil*[13] by *Plena Inclusión Madrid*[14] with which the research group collaborated for the development of the project, whereas pictograms are taken from *ARASAAC*[15]. *ARASAAC* is a company located in Aragona (Spain) which offers a set of

---

[12]https://www.asale.org/academias/real-academia-espanola
[13]Diccionarion fácil (ES) = Easy dictionary (EN)
[14]Plena Inclusión Madrid (ES) = Full Inclusion Madrid
[15]https://arasaac.org/

Figure 7. *Easier* service on a real Spanish text simplifying the word *dictionary* (translated in English).

Figure 8. *Easier* service on a real Spanish text simplifying the word *volumes* (translated in English).

symbols and resources for Augmentative Alternative Communication.

3.3 International Project: *SymWriter2*, Widgit

- structural text simplification

- carried out by an automatic tool

- gratis service

Widgit is an international company based in Warwickshire (England). Founded in 1982, in particular through the use of symbols they designed and developed, they deal with easy and accessible communication for everyone. In fact, on their *LinkedIn* profile [17], they claim "*[...] everybody should have fair and equal access to information.*".

Nowadays, *Widgit* has a database of over 18,000 symbols, all designed to make texts and communication easier, These symbols are used in many of their products - including *SymWriter2*, which we are going to talk about - in order to help users of all ages, abilities and backgrounds who have difficulty.

The company's website [31] is also developed in order to facilitate navigation and communication: moving the cursor over any word on the website, the corresponding icon will appear (Figure 9). In this way, people with difficulty in understanding the term in question, have the opportunity to visualise a representation of it.

But now let's talk about *SymWriter2*, the symbol word processor developed by *Widgit*. It is a program similar to a text-editor that allows you to view the meaning of words as they are typed: special grids are generated around the text, where the corresponding symbols are automatically inserted.

The powerful language processing system developed and used carries out the word-symbol association taking into account multiple grammatical factors: plurals, nouns, articles, prepositions and tenses. Moreover, each user can

exclusion; depending on factors such as the type of disability, where they live and the culture or class to which they belong.

Widgit works with governments ... nies all over the world, providing our symbol ... nd software in 16 languages. See our internat ... ers.

Widgit supports charities that pr ... ucation for children who would otherwise be kept at home and

Figure 9. An example of word-symbol association powered by the *Widgit*'s website.

customise its experience by inserting their own symbols and images within the application. This possibility certainly allows a greater ease of reading, since the symbols in use are familiar more familiar to each user. In addition, each text is spell-checked and can be pronounced by the application, thus offering an additional method of communication.[16]
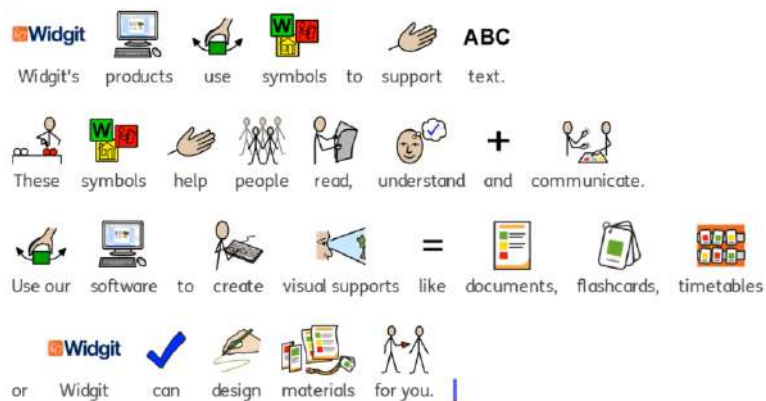

Widgit's products use symbols to support text.

These symbols help people read, understand and communicate.

Use our software to create visual supports like documents, flashcards, timetables

or Widgit can design materials for you.

Figure 10. The way a text is processed and represented by using symbols.

SymWriter2 is not limited to this, in fact it contains many other services, mainly aimed at teaching the act of reading. This teaching section is slightly outside the scope of this paper, so I leave the reader the opportunity to visit and use these tools offered by *SymWriter2*.

3.4   International Project: *Immersive Reader*, Microsoft

- structural text simplification

- carried out by an automatic tool

- free service

Microsoft developed *Immersive Reader* in 2016, which is presented on their website [20] as *'[...] a free tool that uses proven techniques to improve reading for people regardless of their age or ability'*. It aims to help all learners to improve their skills in reading and writing, regardless of their age and ability[17].

It is not an application itself, but a tool that can be activated on many famous services offered by *Microsoft*: *Outlook, Word, Excel, PowerPoint, OneNote, Teams, Publisher, Access*. Myself, since I had never used it, I was amazed that this functionality was present in *Word*. Just indicate that you want to read a text using this service.

There is the possibility to make the text format more accessible: choose the font to use, its color, its size; the spacing; the background color. In addition, *Immersive Reader* automatically recognises the different elements of the text such as verbs, nouns, adjectives, which can be highlighted in different ways to facilitate the presentation of the text. Words can be broken down into syllables, and the text can be made to speak, even at different speeds. In this way, readers in difficulty can, for example, first listen to the text, preparing themselves for a future reading.

Immersive Reader is also offered by *Microsoft* as a free online service [20], where you can easily try it out. In Figure 11, I show you an example of text elaboration: I processed the incipit of the great opera *Novecento* written by the Italian writer Alessandro Baricco, customising the structure of the text and the underlined elements.

---

[16]http://symwriter.auxilia.it/index.php/2016-02-02-13-06-20/introduzione-a-symwriter
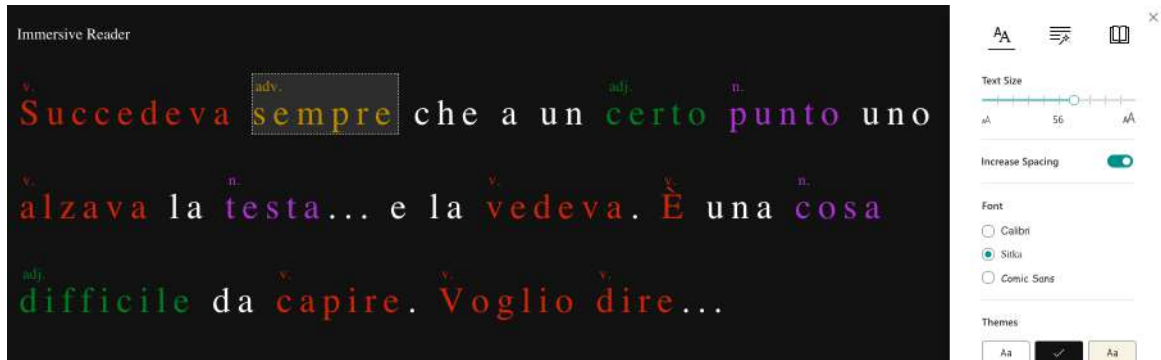[17]https://www.microsoft.com/en-us/education/products/learning-tools

Figure 11. Example of use of *Immersive Reader* on an Italian text.

As well as many applications developed in recent years, *Immersive Reader*'s *Optical Character Recognition* allows you to convert a photo of a text into a digital text, which can be then be simplified for reading.[18]

3.5   Considerations

Through these examples, we have seen that making a text more accessible can be done in many different ways. The projects I have presented are only 4, and each one is completely different from the others in the structure and functions offered. By testing the various features, I realized the need to build my own prototype. A prototype capable of bringing together all the components that, from a theoretical and also a practical point of view, I realized could be of greater help and support during reading.

There is a need to create a new product because such a service is missing for the Italian language. As I will be in charge of the design and construction of the prototype, relying on the material at my disposal, I know I can create something useful. Furthermore, I will be able to customize - not forgetting the main objective of simplicity - the features that most fascinated me to my liking. The project, since it will be at the prototype stage, can then be further customized by experts based on the moment and the use that will be made of it.

---

[18]https://schoolsict.co.uk/what-we-do/office-365/learning-tools/immersive-reader#: :text=Immersive%20Reader%20is%20a%20free, fluency%20for%20English%20language%20learners.

4 Project requirements and analysis

After a first phase of careful reading and analysis of the papers at my disposal on the subject, I moved on to various text simplification systems - such as those in Section 3. Having become familiar with the subject and with the techniques used, I began to think of the key elements to simplify a text, getting a clearer clue of them.

A first idea was to employ artificial intelligence in order to simplify the text. However, these techniques are very advanced even for today's context, they require data sets of original and simplified texts to train models, and the supervision of professional linguists is required. Moreover, Italian language is - according to experts - a very complex language: the lexicon and syntax are very wide and complicated. Therefore, defining a set of patterns on which to implement a simplification - as more easily implemented for example in English - is very complex.

I therefore decided to abandon the path of artificial intelligence, moving towards a more graphic and structural simplification, to which I associated aided communication elements, that are, external elements acting as support for reading.

I started thinking about the most important components to be included in my application. Once they have been defined, I thought about how I could develop them, and what technologies I would have needed. They are presented here below, while the way they are used in the project is explained in the next Section 5.

4.1 Components and technologies

Text-To-Speech

This component is central, because it offers a second communication channel, completely different from the written one, which can accompany or replace it. From many scientific papers - and from a direct discussion assistant Leandro Soares Guedes had with *ProInfirmis* - it is emerged that many people suffering from problems related to reading appreciate this auxiliary mean of communication. The idea is to give the user the possibility to listen to the entered text.

I searched the web for what system I could use for this component, and stumbled upon *NaturalReader*'s *WebReader* [18]. *WebReader* is a text-to-speech tool that must be inserted into your application. You can program it to read a certain content on your page. In addition, some of its features can be customized: the synthesizer's voice and language, the reading speed, and the emphasis of the elements that are read.
The service is free for reading 2'000 pages per day, then there are other paid plans without limitations. Hence, as an applied to software development it's fine, the important thing is that a full version is purchased for commercial use.

There are many other services of this type, the best known of all is certainly the one offered by *Google* [15]. The main problem is that all these services are mostly - and rightfully - paid. *WebReader*, among those that I have tested, is the service that offers the best free trial version.

Synonyms and definitions

In some of the digital systems I studied, synonyms and definitions of words were presented. This operation helps considerably in understanding the word in question. In fact, this is also what happens during translations from original to simplified texts manually carried out by experts. In the presence of complex words, synonyms are evaluated for a possible substitution. In my case it is not a matter of substitution, but of juxtaposition.

I therefore decided to add this component to my work as well. By clicking on a word, the user has the possibility to read synonyms and definitions of it. In order to build it, my idea was to contact an API that provides synonyms and/or definitions given a term. I have long searched the web for an API that provides this type of performance - for example the ones of dictionaries and vocabularies - but without success. In English there are several, one of the most famous is that of *Oxford Dictionaries* [9], but nothing in Italian.

In the absence of APIs, I decided it could be a good idea to locally have my own database of synonyms and definitions. But unfortunately, even finding an Italian database in the correct format is really difficult, then I decided to build it myself. I downloaded a fairly chaotic list of definitions and synonyms [**Bond2012 WordnetWeb**], and processed it with *Python*, to finally get a file containing groups of words by meaning.

Once this database was built, I had to find a way to correctly index it, so that the search for terms could be as fast and efficient as possible. For that purpose, *tweetsearch* - the project I developed last year (2020) during the Information Retrieval course - came to my mind. It consisted in crawling from the web and indexing a collection of tweets, for which I then built a special search engine to navigate such a collection.

Usually, to retrieve data you need to make calls to a backend part, which takes care of managing the interactions with the server and returning the data to the frontend part. But in tweetsearch I used *Solr* [14] - a *Java-based Apache*

search server - and I decided to propose it here again. With *Solr* I indexed the collection of terms synonyms and definitions. This way it is possible to make calls directly to the virtual server on which *Solr* is running. The address of this server depends on the initial configuration of the *Solr* collection. In my case, I decided to use *localhost: 8983*.

By accessing the Solr server from a browser and going to the part dedicated to queries, we can search our indexed terms. You can follow the next example by looking at Figure 12. Let assume we want to search the term *circo*[19]. Following the database structure, we have to specify we are looking for a term, then *term* is our search field. By executing the query, our search results will be shown on the right.
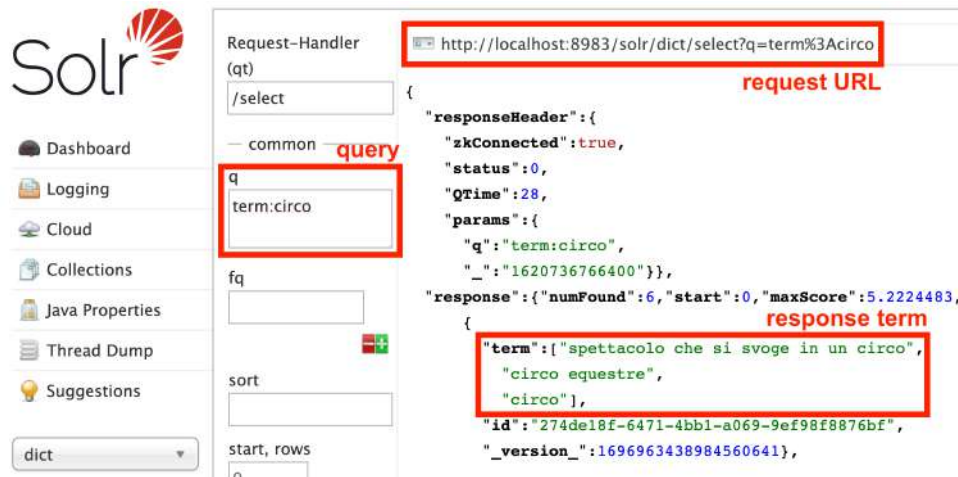

Figure 12. *Solr* browser interface with highlighted the most important elements: query, request URL, and response term.

As you can see, in the upper part a URL is displayed. By making a GET call from a frontend part to this URL, we can achieve the same retrieved results in our web app. The URL is the following:

- http://localhost:8983/solr/dict/select?q=term%3Acirco

By analyzing it, we can identify a structure: the part in orange contains the localhost on which the Solr server is running plus the name of our collection. The part in blue is the dynamic one, which by modifying it allows us to search for all the definitions and synonyms.

Pictograms

One of the elements I cared most about was the possibility for the user to also visualize some of the words in the entered text. This way, in addition to listening and simple reading of the text, a third alternative means of communication was available: the visual one.

I left this component as last as I though this part could be slightly more complex than the others, which turned out to be false. My initial idea was to implement something similar to what I've done for synonyms and definition by using *Solr*. Therefore, I would have needed a collection of images, icons or pictograms to download and index.

I tried this approach, but quickly realized its complications. On the web, there aren't many free image databases to download, which are in the meanwhile also relatively complete. The few that I've found - like *Flaticon* [6] that I have downloaded and tried - contain files in different formats, each of which represents an image. As you can see in Figure 13, the problem is that the names of the images are in English. I could have translated them all manually, but as the collections were made up of many images, this would have been too long or even impossible. I thought of translating them using online services or programs, but the services I identified turned out not to be so efficient.

In agreement with Professor Landoni, we also thought of purchasing the rights to use *SymWriter 2* [31], one of the services presented in Section 3. The service offered is certainly complete, but the license for only one year of use was somehow out of budget.

For this reason, I therefore opted to use a simple API. I needed a service which, given a word, would have returned a image representation of it. In English I found many free, valid and complete APIs, the same is not true for the Italian language. So the problem of translating the terms seemed to be back, until I found a website [4] listing many *AAC* tools. If you are planning to develop something in this area, I strongly suggest you take a look at it since
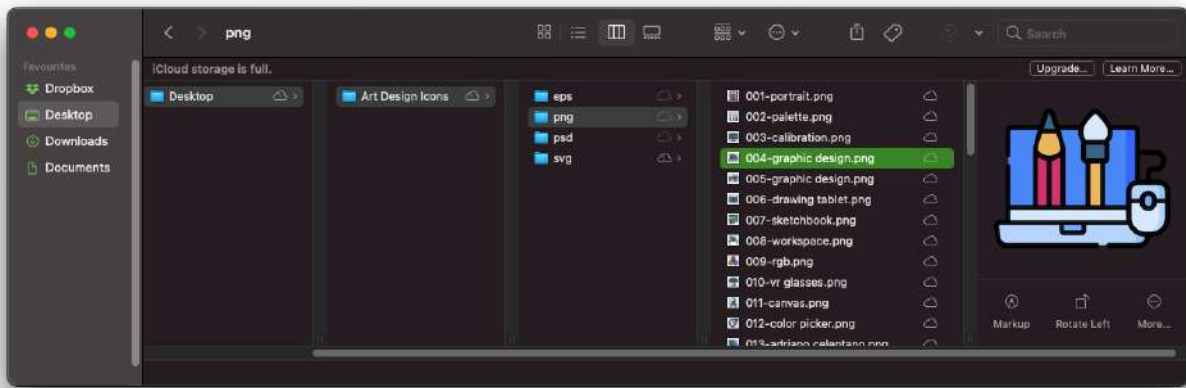
---

[19]circo (IT) = circus (EN)

Figure 13. The way *Flaticon* downloaded folder is organized, as many other services.

is really well done. Here, they featured some copyright free image databases, including *Open Symbols* [7], the one I then actually used. *Open Symbols* offers an API with access to more than 50'000 symbols and icons, also in Italian: perfect for my project!

The API needs an access token for us to be able access images. This token has a validity of 22 hours and must be generated with the following POST request:

- https://www.opensymbols.org/api/v2/token

Therefore, I have arranged the application so that every time it is started, a new access token is requested. After that, with a GET request it is possible to request access to any of the images in the *Open Symbols*' database. Among the headers of the request it is essential to insert the *Authorization* associated with the token we just received. Then, assuming we are interested in visualizing the term *circus*, the anatomy of the address to which we must send the request is the following:

- https://www.opensymbols.org/api/v2/symbols?q=circus&locale=it

The part in orange contains the address of the API to which we are sending the requests. The part in blue is the one that changes depending on the term we are requesting.

Font-size controller

With this feature, the user will be able to change the size of the text at will, so as to have a personalized presentation of the text. The idea is to have a simple slider, this way the user will not directly be in contact with font-size values often reported in pixels, that could be hard to understand.

Font-color controller

The color of the text can also be set as desired. This is to avoid problems in terms of contrast between the background and the color of the font used, with the goal of making the text more visible. This component can be implemented by importing from some library a color picker.

Background-color controller

The reasoning is the same but applied to the background: certain combinations of font and background colors might be more appreciated by a certain type of people.

But how to connect all the pieces together? *tweetsearch* - the search engine I built last year for the Information Retrieval course - was a single-page web application. I found the single-page structure interesting, and I therefore decided to propose it again in this project. But I had to decide which system to use to create the user interface. Since in the internship offered - again last year - in the Field Project Atelier course, the company *Zucchetti SA Switzerland*

asked me to build a web app using *Vue.js* [30], I decided to use the same system here, so as to increase my familiarity with it. *Vue.js* is a *Javascript* framework for creating user interfaces and single- or multiple-page applications, exactly what I need!

5 Project design (top-down)

When I started thinking about the project, I immediately thought about the people to whom it is mainly dedicated, their educators and families. I absolutely did not want my work to be an obstacle to learning, but a help. So, I decided for a simple, intuitive and easy to use design: generally, the interface should not contain too many elements, capable of disturbing the user's attention.

The application is built so to have a single page with two different views: we will call them *Home* and *Simpler*. Each of these two pages is presented in detail in the following sections.
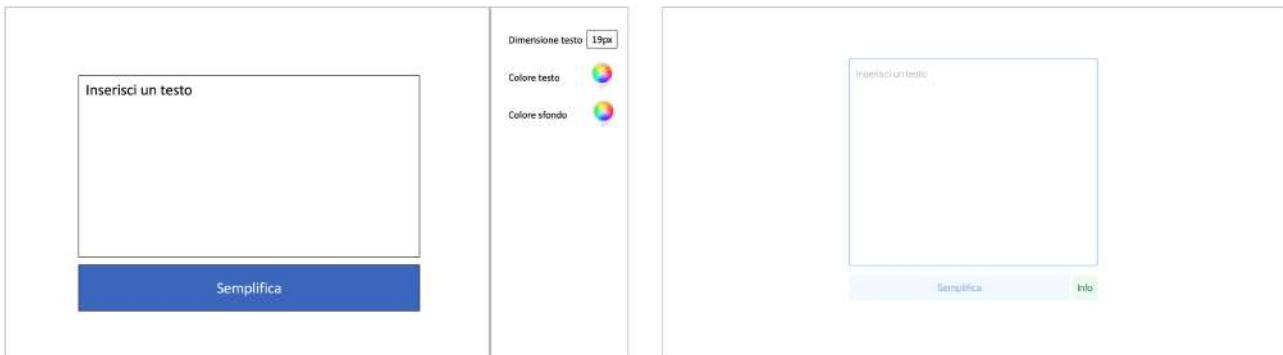
5.1 Home page

Figure 14. Home page interface: from the mock-up (left) to the final result (right).

This is also the main page of the web application, the one I started designing first. Here, there must be a **text area** where the user can write or paste the text to be simplified. Once the text has been entered, the user must be able to immediately simplify it: this can be done by clicking on a *Semplifica*[20] **button**. After clicking on the button, the user will visualize a simplified version of the text: more accessible and much easier to read. I will describe it in the following Subsection 5.2. The two components in *Home* had to be central, so I decided to place them in the center of the page, so as to be of greater impact.

Figure 14 shows the mock-up of the *Home* interface I made before starting so to have a reference, and the *Home* interface of once completed.

5.2 Simpler page

Figure 15. Simpler page interface: from the mock-up (left) to the final result (right).

Here, the user visualizes the simplified version of the submitted text. The access to this page is guaranteed only after the submission of the text, that cannot be further modified. The view opens with a pop-up column on both left and right sides of the screen. This helps to understand that the topics included in the first and second columns are of a different nature. This division is also dictated by order and convenience in organizing a simple interface. You

---

[20]Semplifica (IT) = Simplify (EN)

can have an idea of the view by looking at Figure 15.

The left column is titled *Aiuto*[21]. Its task is to offer a clearer view of the term currently being investigated by the user. Indeed, by clicking on terms, their **definitions/synonyms** and **pictograms** will appear (clearly if they are present in the databases). In many cases, the database that I built for definitions/synonyms returns multiple definitions/synonyms for a single term, since meanings also depend on the context a term is located. That's exactly why I decided to present that component in a carousel format, that is, a kind of slideshow for cycling them in confined spaces. There are three placeholders - *Parola selezionata*, *Immagine*, *Definizione/Sinonimo*[22] - that mark the possibility of obtaining these three indicated services by clicking on a term.

While the left column - helping to understand the text - is more static and related to a single term, the one on the right, called *Personalizza*[23], is more interactive and related to the whole text. The purpose of the first three components is to give the user the possibility to format the text according to their needs. He will be able to modify the **color of the background, and color and size of the font**. The idea was to build them with, respectively, two color-pickers and a size text-field. As you can see in Figure 15, the text-field has been replaced during the project development with a slider: more intuitive and easy to use. Another key element present in this column is the ***Attiva audio***[24] **button**, by clicking on which we are able to listen the just processed text.

An important note is that all these elements are only useful in the this page, since the role of the *Home* page is restricted to receive the text to be processed. Hence, these elements must only be visible in this page. For this reason, I positioned them so to appear in the exact moment an input text is submitted to be simplified, without being disturbing.

The last element that I want to describe is the ***Semplifica un altro testo***[25] **button**. When the user clicks on it, the current simplified text will be erased and the user will be brought to the *Home* page again. Here, he will have the chance to effectively undergo another text to the simplification process.

5.3   Further details

I want to point out that to make everything even more accessible and intuitive, I tried to guide the use of the application. After getting a stable version of the application, I had the idea of putting in some mini-instructions. On both pages, on the right of the main blue button - *Semplifica* in Home and *Semplifica un altro testo* in Simpler - I have inserted the information ***Info*** **button**. By clicking on this button a small pop-up will appear which briefly describes how to use the current page. This tool is depicted in Figure 16.

Figure 16. By clicking on the *Info* button, Italian mini-instructions on how to use the current interface will appear.

In this way, should users have difficulty in understanding how to navigate within the application, they can view a small guide: schematic, short and simple.

---

[21]Aiuto (IT) = Help (EN)
[22]Parola selezionata (IT) = Selected word (EN); Immagine (IT) = Picture (EN); Definizione/Sinonimo (IT) = Definition/Synonym (EN)
[23]Personalizza (IT) = Customize (EN)
[24]Attiva audio (IT) = Turn on audio (EN)
[25]Semplifica un altro testo (IT) = Simplify another text (EN)

6 Implementation issues (bottom-up)

After having structured the collection of synonyms/definitions in order to be indexed at best through *Solr*, I ran the *Solr* server and tried to contact it from the frontend: every request I was sending from my localhost was rejected by the *Solr* due to the **CORS policies**. *CORS policies* are security mechanisms that allow to limit the request for resources from another external domain [8]. Then, after a lot of research and attempts, I've ended up in modifying the *Solr* configurations *web.xml* file. This file is located at path *solr- 8.6.2/server/solr-webapp/webapp/WEB-INF* in the *Solr* directory and is responsible for the filters and servlets used by the application [22].

I thought I would no longer encounter problems of this type, but instead the exact same problem arose when working with the API for images: every request sent to *Open Symbols* was blocked. The solution implemented earlier could not be used in this case. Previously, I directly modified the backend part to allow external requests to be accepted. Now, dealing with a public API, I didn't have access to the source code, and I couldn't modify it as I wanted.

After countless searches online, I realized that one of the only way to get my requests accepted and executed was to use a *proxy server*. A proxy is a type of server that acts as an intermediary for requests from a client - in this case my *localhost* - looking for resources on another server - that of *Open Symbols* [33]. I send the requests to the proxy, which takes care of forwarding them to the target server and getting a response. This response is then passed to me, the original sender of the request. My *localhost* did not have the necessary permissions to receive a positive response from the API, but the proxy I used did. Figure 17 helps in the interpretation of the above mentioned mechanism.
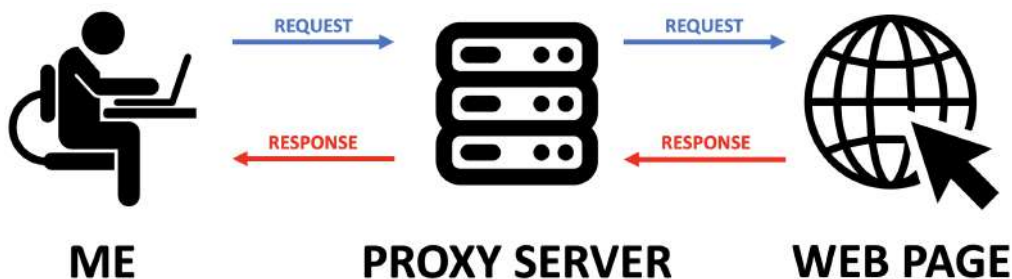
Figure 17. The functioning of a proxy server with the 3 main actors: user (me), proxy server and desired web resource.

So, I added the *cors-proxy-server* package [2] to my application modules, which allows to generate a proxy server at *0.0.0.0:9090* address. The anatomy of the requests sent to the API of *Open Symbols* has therefore undergone a change. This is the final version:

- http://localhost:9090/https://www.opensymbols.org/api/v2/symbols?q=circus&locale=it

As you can see, now there are kind of two addresses: in orange I depicted the address of the proxy server that I added, while in blue you can see the same string as before.

7 Application testing: experts and users evaluation

From the beginning of the project, the idea was to create a finished and working application to be tested by a group of users. In this way, we would also have obtained feedback from possible direct future users of the service.

The application is - as mentioned in the Introduction [1] - aimed at anyone who wants to benefit from it, but with a particular regard for people with intellectual disabilities ID. Therefore, as is done in some cases, the target user group on which to test the application did not have to be a random and representative sample of the population, but precisely that of ID people. This, according to the *quasi-experiment* pattern, in which the study sample is carefully selected. In fact, many of the works cited in this report make use of this technique and obtain feedback in line with their goals. I would like to remind you that the simplifications that are implemented do not act as an obstacle for categories of people not affected by ID. Our goal was also to simulate an inclusive research process.

Unfortunately, finding this user group was not possible in time. This is also due to the COVID-19 pandemic crisis that is afflicting the world at the time I'm writing (May 2021). But we still plan to run this user assessment when the situation is calmer. This is also described in the next section [8].

For the time being, on May 25 (2021) we have been able to have a video call with Marilina Mastrogiuseppe (MM), which is a postdoctoral researcher at the University of Trento. MM, together with its research groups, has also published some of the papers that I analyzed during the production of this work [11, 12, 19].

The aim of the meeting with the expert was to get a clearer idea on the experimental procedures to follow in presenting the work both to the experts and to the final users of the application. We wanted also to obtain a high level evaluation regarding my prototype and the *AAC* techniques contained therein. In fact, the expert spoke about the structure of my prototype, giving advice and possible improvements.

The simplicity of the interface was appreciated, as it makes everything more elegant and immediately usable. Regarding it, in many fields and in many components of the product, it is certainly better to replace the written language with representative icons. An illustrative example would be to replace the word *Definizione/Sinonimo*[26] on the SIMPLER page with a dictionary icon. By doing so, particularly people with intellectual disabilities - who have limited executive functions during the act of reading - can understand the various functions more easily. In fact, images and icons stimulate cognitive resources in the subject, and this helps to generate a mental representation of the various functions present.

These limited executive functions during reading are in parallel aided by the text-to-speech component. This component has been greatly appreciated, as it provides an alternative way of communication to simply reading printed characters.

The specialist also recommended removing the ability to click on each word. She mentioned that it could be useful to make only low-frequency Italian words clickable (which are the least known, and therefore less understandable). But, as already mentioned, my prototype does not perform any syntactic or grammatical analysis. And to implement this feature, a linguistic search on a larger or smaller dataset of documents should be performed to determine the low-frequency terms. The suggested solution is however very close to the one proposed by the HULAT research team with EASY. In fact, in their application only the terms considered complex were highlighted and made interactive.

From the discussion with MM we also defined how to carry out the evaluation with the experts. The experimental pattern we use is that of the **cognitive walk-through**. This technique consists in - I cite the authoritative source *Usability Professionals' Association* (UPA) [29] - *'one or more evaluators work through a series of tasks and ask a set of questions from the perspective of the user'*. The goal of the cognitive walk-through is to understand the learning ability of the system by end users. We therefore decided to define a series of tasks regarding the application that I designed to be executed. Each of them wants to test a different part of the system: we want to investigate the basic simplification of a text, automatic reading, and the customization of the page structure. These are visible in the first column of Figure 18. For each of these tasks, the goal is to collect as much information as possible about their implementation. In fact, we will record a whole series of information, mainly relating to the difficulty and success of completion. Any comment will also be accepted and promptly recorded. These questions that we have decided to ask are visible in the first row of Figure 18.

Later, taking advantage of the meeting with an expert on the subject like MM, we asked for advice on how to carry out a user evaluation. We cannot simply define tasks and ask users to perform them, since, in this particular

---

[26]Definizione/Sinonimo (IT) = Definition/Synonym (EN)

	TASK	Understandable	Correct execution	Task completed	Needed help	Comments
1) Find information for a complex term	Enter a complex term in the input window					
	Simplify					
	Click on the complex term					
	Report what obtained on the left section					
2) Use the text-to-speech function to listen to an entered text	Enter a text in the input window					
	Simplify					
	Click the text-to-speech button					
	Report if what you hear relates to the entered text					
3) Customize the text structure	Enter a text in the input window					
	Simplify					
	Using the features on the right section, change font size, font color, background color					
	Report any visible changes in the text entered					

Figure 18. The cognitive walk-through to be submitted to the experts.

case, target users are people with ID, who, without a specific experimental setup, would struggle somewhat. We want to know if, by providing the right instructions, my prototype can facilitate the reading of an Italian text. The specialist is often in contact with research groups and people with ID and knows how to best approach them, then we asked her for her opinion.

First of all, it is useful for research purposes to collect information about the subjects of the experiment. Some more classical information, such as age, gender, and level of education; others more targeted, such as the MT tests. MT tests, developed by the Institute of Psychology of Padua, are a series of tests that allow to evaluate the reading skills in speed, correctness and comprehension in young people from the first year of primary school up to the two years of secondary school [10].

These tests fit very well in our scenario, since the age of this population group - ranging from 6 to 14 years - corresponds to the mental age of the ID people, which in average is 10 years old. Through MT tests, we can define the ability of the experimental subject to understand the written text. All these collected data can be used at the end of the experiment, in order to generate clusters based on the different aspects and abilities of people.

MM advised us to carefully choose the texts to be submitted to the ID users. In fact, it is preferable to select texts already known by these people. In this way, having already a certain mastery of the subject (or even a minimum knowledge of it), the subject knows a priori what he could expect from the content of the text. This aspect helps in the contextualization of the submitted passage, which is linked to a cognitive experience already lived. All these elements make the use of the passage more understandable to the subject, and generates greater security in them, inhibiting any level of anxiety.

Mastrogiuseppe continues stating that by putting ourselves in the shoes of a person with ID, we could be confused in front of the interface. This is because we may not grasp the usability of the application, a very important aspect because it allows us to understand and convey the reason why we should use this application. To explain usability to ID people, it would be better to produce a demo of the application - perhaps a video -, in which the various simplification functions offered are used. To stimulate the understanding of the subjects, MM advised to insert explanatory smilies within this demo. For example, after entering a text and clicking on *Semplifica*[27], a happy face may appear on the screen, as if to indicate that the simplification act has been completed. At this point, a face with obvious vision problems may appear, indicating that the size of the text needs to be adjusted.

---

[27]Semplifica (IT) = Simplify (EN)

All these ideas presented by the specialist can be taken into consideration to create a scripted or semi-scripted user evaluation.

8 Conclusions and future work or possible developments

The possible developments of the project that I would suggest have to do in particular with the technological solution that I have built. In fact, all the components used have - yes - been chosen with care, but with the aim of not having to spend in order to use them.

The database of synonyms/definitions is the best list I found online in Italian. I then had to process it with my Python script and connect with a backend part (*Solr*). It is one of the best lists of terms, but some are absent, while among those present some are not good at all and can also be confusing. Therefore, in case you want to raise the project to a production level, rather than just development, I highly recommend using other databases, most likely for a fee. This can be built by buying a dictionary list, or by using a special API that allows you to receive synonyms/definitions once the request is sent.

The same reasoning applies to the image database. Many words have no representation, while others are represented but in a misleading way. For this reason, I recommend opting for other existing services. One of the many is *Widgit* that I introduced in Section 3: 15,000 images correctly associated with terms. Otherwise, if the use of the project is restricted to a particular branch, you could collaborate with a service that produces only the small number of images needed, in order to have your own images. Having the images in your own database is important. In this way you rely exclusively on yourselves and on your abilities, without delegating to third parties, that could suddenly disappear or have problems without your knowledge.

A final component that could be improved is the text-to-speech function. When activated, the voice sounds too robotic and is not nice to hear. This is a problem because such a function is supposed to offer an alternative communication channel, but in doing so it discredits it. There are countless functions of this type for a fee, first of all the one offered by Google and already mentioned in Subsection 4.1.

A possible future development of this project could also be to explore AI techniques and elaborate some functions. For example, in addition to simplification and improved accessibility already implemented, the text could be processed and reproduced in a syntactically simpler version. Being able to choose between some levels of simplification before processing the text could also be an innovative and winning choice. Another idea is to recognize the different elements of the text - such as nouns, verbs, adjectives, etc. - in order to present them to the user in a different way, highlighting some because more important.

All this material that I have collected and analyzed - and that you can find in Section 8 - has allowed me to have a broader view of this subject which many have ignored. I was also able to design and finish a fully functional project. For this reason, with the collaboration of my supervisor Professor Monica Landoni and the doctoral assistant Leandro Soares Guedes, we have decided to produce and possibly publish a scientific paper based on this work.

The paper must be written in a particular format, that of this work doesn't fit because there are some elements too many, while others are missing entirely. We want the paper to be as complete as possible, so we aim to add a two sections dedicated to the evaluation of the application: a discussion with experts in the field and a session with users with real difficulties. About the testing part with real users, we are waiting for the pandemic situation to improve, so that we can identify a group of people to try the application, and collect all their feedbacks.

The conferences we can send to are many, but we have decided to opt for *Conference on Human Factors in Computing Systems* (*CHI*) by *ACM*. *CHI* is a series of academic conferences related to the interaction between human and computer, also considered one of the top-ranked conferences in computer science. By mutual agreement, in the human-computer interaction field, *CHI* is indicated as the most prestigious. For example, this year it was supposed to take place in Japan, but due to the pandemic, it was held in digital form. The sponsors are of the caliber of *Google*, *Apple*, *IBM*, and the participants were thousands. Next year's conference is scheduled for spring 2022, and we are organizing ourselves to be able to draft a paper that respects the rules and CHI format [1].

I am very happy to have pursued all the objectives that I had initially set myself, respecting the schedule of times that I had built. The prototype is functional and is based on a series of techniques proven and suggested by the scientific community and years of research in this regard.

References

[1] ACM. *CHI: Conference on Human Factors in Computing Systems*. `https://dl.acm.org/conference/chi`. Accessed: 2021-17-05.

[2] Antonio Aguilar. *npm - Simple CORS proxy server*. `https://www.npmjs.com/package/cors-proxy-server`. Accessed: 2021-17-05.

[3] UN General Assembly. *Convention on the Rights of Persons with Disabilities*. A/RES/61/106, Annex I. UN General Assembly, 2006. URL: `https://www.un.org/development/desa/disabilities/convention-on-the-rights-of-persons-with-disabilities/convention-on-the-rights-of-persons-with-disabilities-2.html`.

[4] Open Assistive. *Open Assistive - A curated list of awesome Assistive Technology frameworks and tools to help you develop your AT tool/system*. `https://openassistive.org/awesome-assistivetech/#speech-synthesis`. Accessed: 2021-17-05.

[5] American Psychiatric Association. *What is Intellectual Disability?* `https://www.psychiatry.org/patients-families/intellectual-disability/what-is-intellectual-disability`. Accessed: 2021-05-05. 2017.

[6] Freepik Company. *Flaticon - Free vector icons*. `https://www.flaticon.com/`. Accessed: 2021-10-05.

[7] CoughDrop. *Open Symbols - Open-licensed communication symbols for everyone*. `https://www.opensymbols.org/`. Accessed: 2021-17-05.

[8] Mozilla Developer. *Cross-Origin Resource Sharing (CORS)*. `https://developer.mozilla.org/en-US/docs/Web/HTTP/CORS`. Accessed: 2021-17-05.

[9] Oxford Dictionaries. *Oxford Dictionaries API - Enhance your app with our world-renowned dictionary data*. `https://developer.oxforddictionaries.com/`. Accessed: 2021-17-05.

[10] Giunti Edu. *Il progetto MT per la scuola*. `https://www.giuntios.it/_tdz/@media_manager/9500359/`. Accessed: 2021-26-05.

[11] Bortolotti Elena and Mastrogiuseppe Marilina. "Accessibilità ai siti culturali per la disabilità intellettiva: metodologie, tecnologie e processi di adattamento". In: *NUOVA SECONDARIA* 37(2019) (2019), pp. 158–170.

[12] Bortolotti Elena, Mastrogiuseppe Marilina, and Span Stefania. "Removing barriers to communication for people with Intellectual Disabilities in cultural spaces. The analysis of processes for simplifying cultural contents". 2021.

[13] Inclusion Europe and Anffas Onlus. *Informazioni per tutti - Linee guida europee per rendere l'informazione facile da leggere e da capire per tutti*. Inclusion Europe and Anffas Onlus, 2009. ISBN: 2-87460-131-4.

[14] The Apache Software Foundation. *Solr - Solr is the popular, blazing-fast, open source enterprise search platform built on Apache Lucene*. `https://solr.apache.org/`. Accessed: 2021-17-05.

[15] Google. *Text-to-Speech - Convert text into natural-sounding speech using an API powered by Google's AI technologies*. `https://cloud.google.com/text-to-speech?hl=en`. Accessed: 2021-17-05.

[16] Human Language and Accessibility Technologies (HULAT). *Easier - Tools to simplify texts in spanish*. `http://163.117.129.208:8080/project`. Accessed: 2021-10-05.

[17] Linkedin. *Widgit — Linkedin*. Accessed: 2021-17-05. 2021. URL: `%5Curl%7Bhttps://www.linkedin.com/company/widgit-software/about/%7D`.

[18] NaturalSoft Ltd. *WebReader - Let the website talk for you*. `https://www.naturalreaders.com/widget.html`. Accessed: 2021-10-05.

[19] Marilina Mastrogiuseppe, Stefania Span, and Elena Bortolotti. "Improving accessibility to cultural heritage for people with Intellectual Disabilities: A tool for observing the obstacles and facilitators for the access to knowledge". In: *Alter* 15.2 (2021), pp. 113–123. ISSN: 1875-0672. DOI: `https://doi.org/10.1016/j.alter.2020.06.016`. URL: `https://www.sciencedirect.com/science/article/pii/S1875067220300651`.

[20] Microsoft. *Immersive Reader - Microsoft Learning Tools*. `https://www.onenote.com/learningtools`. Accessed: 2021-10-05.

[21] Sergiu Nisioi et al. "Exploring Neural Text Simplification Models". In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Vancouver, Canada: Association for Computational Linguistics, July 2017, pp. 85–91. DOI: `10.18653/v1/P17-2014`. URL: `https://www.aclweb.org/anthology/P17-2014`.

[22] Oracle. *web.xml File - ATG Web Services and Integration Framework Guide*. `https://docs.oracle.com/cd/E26180_01/Platform.94/ATGIntFrameGuide/html/s0204webxmlfile01.html`. Accessed: 2021-16-05.

[23] CAST (Understood partner). *Universal Design for Learning (UDL): A teacher's guide*. `https://www.understood.org/en/school-learning/for-educators/universal-design-for-learning/understanding-universal-design-for-learning`. Accessed: 2021-16-05.

[24] Horacio Saggion and Graeme Hirst. *Automatic Text Simplification*. Morgan amp; Claypool Publishers, 2017. ISBN: 1627058680.

[25] Horacio Saggion and Graeme Hirst. "Readability and Text Simplification". In: *Automatic Text Simplification*. Morgan amp; Claypool Publishers, 2017, pp. 7–19. ISBN: 1627058680.

[26] Hubmann Stefania. "La lingua facile". In: *Azione* (2019). Accessed: 2021-05-05. URL: `https://www.azione.ch/societa/dettaglio/articolo/la-lingua-facile.html`.

[27] Mari Carmen Suárez-Figueroa et al. "Towards the Assessment of Easy-to-Read Guidelines Using Artificial Intelligence Techniques". In: *ICCHP (1)*. 2020, pp. 74–82. URL: `https://doi.org/10.1007/978-3-030-58796-3_10`.

[28] Sara Tonelli, Ke Tran Manh, and Emanuele Pianta. "Making Readability Indices Readable". In: *Proceedings of the First Workshop on Predicting and Improving Text Readability for target reader populations*. Montréal, Canada: Association for Computational Linguistics, June 2012, pp. 40–48. URL: `https://www.aclweb.org/anthology/W12-2206`.

[29] Usability Professionals Association UPA. *Cognitive Walkthrough*. `https://www.usabilitybok.org/cognitive-walkthrough#:~:text=The%20cognitive%20walkthrough%20is%20a,for%20new%20or%20infrequent%20users.`. Accessed: 2021-02-06.

[30] Vue.js. *Vue.js - The Progressive JavaScript Framework*. `https://vuejs.org/`. Accessed: 2021-17-05.

[31] Widgit. *SymWriter 2 - Easy symbol writing for everyone*. `https://www.widgit.com/products/symwriter/index.htm`. Accessed: 2021-10-05.

[32] Wikipedia. *Indice Gulpease — Wikipedia, The free encyclopedia*. Accessed: 2021-17-05. 2021. URL: `%5Curl%7Bhttps://it.wikipedia.org/wiki/Indice_Gulpease%7D`.

[33] Wikipedia. *Proxy server — Wikipedia, The free encyclopedia*. Accessed: 2021-17-05. 2021. URL: `%5Curl%7Bhttps://en.wikipedia.org/wiki/Proxy_server%7D`.

[34] Wikipedia. *Text Simplification — Wikipedia, The free encyclopedia*. Accessed: 2021-17-05. 2021. URL: `%5Curl%7Bhttps://en.wikipedia.org/wiki/Text_simplification%7D`.