Università
Ca'Foscari
Venezia

Master's Degree
in Computer Science

Final Thesis

# Auditor: a webpage accessibility validator for people with autism spectrum disorders

**Supervisors**
Ch. Prof. Flaminia Luccio
Ch. Prof. Claudio Lucchese

**Candidate**
Margherita Rossi
Matriculation number 842790

**Academic Year**
2018/2019

# Contents

# Abstract

During the last years some standards have been established in order to make the Web accessible for anyone. Following these guidelines, different webpages validators have been developed, to help Web programmers and users understanding if a page is accessible by people with disabilities. Many disabilities have been considered, e.g., auditory, cognitive, neurobiological, physical, speech and visual disabilities, and we can find many validators around the Web, but there is no specific validator for people with Autism Spectrum Disorders (ASD). The problem of ASD is that it involves different types of difficulties, e.g., in text and images understanding, as well as problems looking at animated objects or listening to specific sounds, etc., it is thus difficult to develop a general validator. Presently, there are many theoretical guidelines for the development of websites for people with ASD, but no specific validator exists. Following the existing guidelines we have developed a validator which can be used by users to understand if a page is accessible for people with ASD. Moreover, it indicates to website developers which are the critical issues that make the website non-accessible.

# Introduction

This thesis is the result of research and careful study in the field of Web Accessibility and Usability, particularly for people with Autism Spectrum Disorder (ASD).

In the Web it is quite easy to find support tools for various types of disabilities, from those for deafness to those for blindness, just as it is easy to find validators that clarify to developers and to end users whether a certain content is accessible or not for people with these disabilities. However, it is very difficult to find tools to understand if a content is accessible to people with ASD, because, as we will discuss in the next chapters, this type of disability cannot be uniquely characterized but includes different types of features and limitations.

The objective of this thesis is to create a Web pages validator tool that can be used both by developers or by teachers/health operators/relatives in order to understand if and how much a Web content is accessible to people with ASD. The tool is based on some guidelines provided by Web Accessibility Organizations, on those discussed in different research papers, as well as on personal studies and reflections that combine transdisciplinary skills in different areas. The tool is very general, so to support as many types of disabilities as possible within the autistic spectrum. We think that it can be a good starting point for future studies dedicated to this topic.

The thesis is organized as follows. In the first chapter we present the Autism Spectrum Disorder describing what characterizes this disability and some existing interventions.

In the second chapter we discuss Web Accessibility in general by describing in detail the guidelines provided by the Standards Organizations such as the World Wide Web Consortium (W3C), and the ones proposed by some international government groups. We also present dedicated guidelines for people with ASD based on different research papers.

In the third chapter we discuss the theme of Natural Language Processing: we provide some definitions to make the reader understand how artificial intelligence can compute the readability of a written text, i.e. saying if a text is simple or difficult to understand.

The fourth chapter presents the main validation tools that can be found online, which form the basis of our final work.

In the fifth chapter we describe the core of the thesis, i.e. the Auditor tool, by describing its structure and by explaining the stylistic choices, as well as the solutions adopted to try to face some of the main barriers that people with ASD can find while navigating the Web.

To conclude, we want to highlight that this thesis covers different topics starting from computer science, and moving to psychology and linguistics. This demonstrates the transversality of the subject, its complexity and its importance for a full inclusion of a person with ASD.

# Chapter 1

# Autism Spectrum Disorder

In this chapter we will present an overview on the Autism Spectrum Disorder. We will describe the disorder, the terminology, the symptoms, some possible causes and statistical data. We will describe the main difficulties autistic people encounter when consulting a website. We will also illustrate some techniques and therapies to treat it.

## 1.1 Terminology

The *Autism Spectrum Disorder (ASD)*, is a neurological-developmental disorder which appears starting from the first three years of life. The term *autism* comes from the Greek αὐτός, which means «self»: this is the best way to summarize the pathology, which makes affected people appear like they are closed into themselves, feeling the reality in a different way from *Typically Developing (TD)* people.

In [11], the authors propose two different ways of looking at autism that depends on the approach one may have to the disorder: the person-first way and the disability-first way. From a terminology point of view, in the person-first way approach the words used to indicate a person with this disorder are "people with autism" or "people with Autism Spectrum Disorder". In disability-first way, the term used is "autistic people". Professionals usually adopt the person-first approach, while many autism community members prefer the second one: the reason is that the second approach frees autism from any negative meaning and describes the disorder as a simple condition. The scientific community instead agrees to define autistic people as *Neurodifferent*, while non-autistic people as *Typically Developing people* [37]: we will keep this duality.

When we talk about Autism Spectrum Disorders, we are grouping the different shades of this diagnosis, as we can see in Figure 1.1. Thus, autism cannot always be brought back to the same disabilities, but it may cause different symptoms and disability levels: it may affect speech and non-verbal communication, social interactions, physical abilities; it may also cause visual and auditory disabilities. In order to understand people with ASD, we need to know that they perceive reality in a different way, and this has an effect on all their attempts of communicating with us. This way of being, causes impossibility to understand their feelings and needs by TD people, and this often leads them to frustration, which can be vented with screams or even a greater closure, repetitive movements, etc..


Figure 1.1: ASD diagnosis components.

The following is a list which contains examples of some possible behaviour an autistic child or an adult might have [42]:

- not be responding to his name;
- avoid eye contact;
- not be smiling when someone smiles at him;
- getting very upset for some taste, smell or sound;
- do repetitive movements or repeat the same sentence;
- not be talking as much as the others;

- not seeming to understand others thoughts or feelings, and have difficulty to express his own;
- have strict daily routine, and be upset for changes;
- be taking things very literally;
- be in a state of anxiety in social situations;
- seem blunt, rude or not interested, and not understand social rules;
- get too close to other people, or feel annoyed when touched;
- notice small details, patterns, particular smells or sounds that others do not;
- need a careful planning of things.

Each person with ASD is different and can show one or more of the above non-standard behaviours.

## 1.2 The causes of Autism Spectrum Disorder

The causes of this disorder are unknown, but the scientific community states that the problem is given by a combination of genetic, neuro-biological and psycho-environmental causes [37]. Since, as already said, autism is a condition which presents different levels of disability, not all these symptoms can be found on a single person and this makes the diagnosis more difficult.

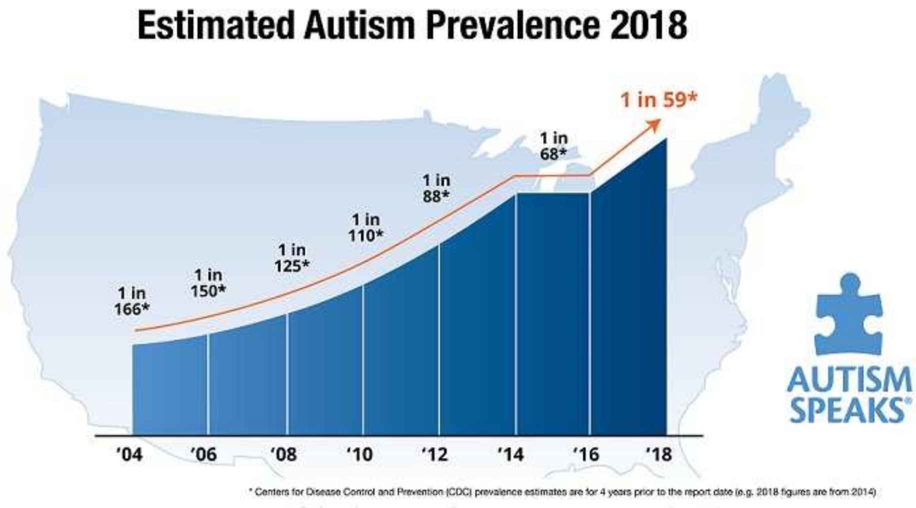


Figure 1.2: Statistic of autistic people in the last year in USA [31].

Thus, in the past many people with autism have been classified as mentally retarded, basing the classification on some tests that were not measured for this specific disorder, altering the statistics. The data we have right now are more accurate, since the diagnostic methods have become better and there is greater awareness on the argument.

During the last years there has been an increase in the diagnosis of autism, but it is not clear if this is real or if it is only due to the fact that the medical community is more informed and prepared to this: some data about the incidence in the U.S.A are shown in Figure 1.2.

In the last years some programs have been developed in order to better understand and monitoring autism and to increase the knowledge and awareness on the topic. Examples are the *Autism Spectrum Disorders in the European Union (ASDEU)* program [30], and the *Autism and Developmental*

*Disabilities Monitoring (ADDM) Network* in the United States [32]. Note that both programs have reported an increase of the number of cases in the last 30 years, with a higher incidence in males.

## 1.3 Sensitivity in autism

Recently, many studies have sought to bring together the most common features of autism spectrum disorders, analysing in particular the sensory experiences in adults and children affected by ASD. These studies have confirmed the already well-known opinion that autistic people are not only particularly sensitive to sounds, images, colors, but also to smell and touch [2]. Since we want to address the problems that could be encountered when consulting a Web content, we will focus on the acoustic and visual stimuli that can cause problems to an autistic person, leaving aside the other two aspects which are not so central in this context.

**Auditory Sensitivity.** When we talk about auditory stimuli, we refer to all the sounds, the frequencies, the volume with which they are reproduced. Some studies and information campaigns have shown that autistic people are generally bothered by loud sounds. [37]
This results in an extreme difficulty in isolating the sound from the rest of the reality, often leading the person to reactions such as psychological isolation, screams, and other behaviors. This is true for every kind of sound except for certain types of music, which in some cases have proven to provide

relaxation: indeed, music can help them to communicate in alternative ways, and for this reason the experts suggest music therapy (see section 1.4) [26]. As we know, the audible musical sounds for man are in the range between 16/20 Hertz and 16,000 / 20,000 Hertz, with a progressive lowering in the ability of hearing them which begins approximately at the age of twenty. Obviously, only a little mention can be dedicated to sounds, since the discussion about this topic is complex and it would need more basic knowledge and time. There are basically three components to sound:

- the height, which is given by the number of vibrations per second and which is measured in Hertz;
- the intensity, which is given by the amplitude of each vibration and is measured in decibels (which must always be associated with the distance from the source) or in Watts;
- the timbre, which is given by the sum of all the components of each sound (nature cannot provide pure sounds: all the sounds are composed by the fundamental sound and the sum of the harmonic sounds), multiple of the fundamental frequency.

Working on the human ear, scholars have realized that the ability to “hear” each sound differs mainly according to the frequency, and the intensity with which each sound is emitted: the central sounds are easier to be heard rather than the high-pitched and low-pitched ones. A very low-pitched sound and, above all, a very acute one needs more strength to be heard easily. Scholars have identified, in addition to the threshold of audibility, the pain threshold: if a sound reaches it, this can actually annoy the listener; a graphical representation of these thresholds can be seen in Figures 1.3 and 1.4. There are also rare cases where the frequency and intensity interact with each other, but they can be neglected here.

According to international agreements, the central A key of the piano is 440 Hertz; all the lower A's are obtained by dividing this quantity by two, the higher ones by multiplying the same figure by two. Since the Hertz is a logarithmic measure, all the other values must be considered with dedicated calculations.

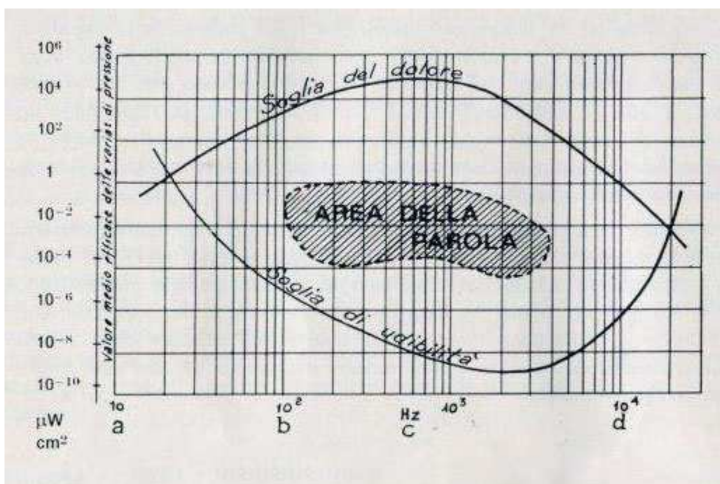


Figure 1.3: Average normal audiogram (Audibility area)[19]

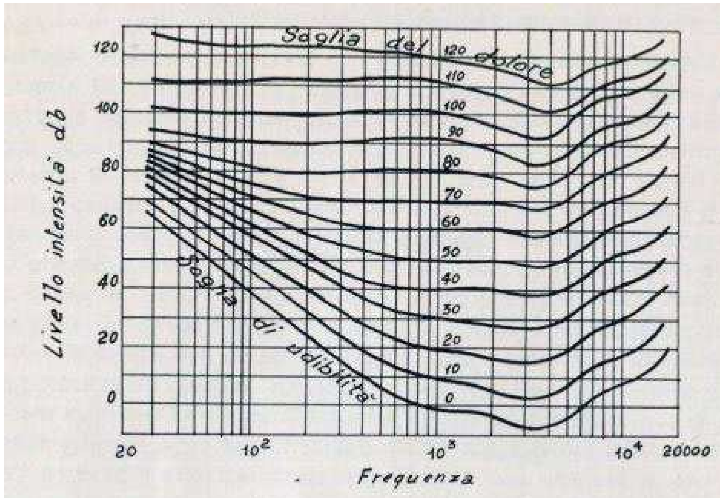


Figure 1.4: Isofone curves[19]

However, beyond the physical dimensions of the sounds, there are still many unresolved fundamental questions about the relationship between autism and sounds. When can a musical composition be annoying to people affected by autism? The Tomatis Method illustrated in section 1.4 highlighted that Mozart's Music is appreciated in certain situations, but what about autism? Does this apply to every author from any period? Can sounds used on the internet negatively affect the well-being of the listener? We all know about the presence of what is called white sound, which is the sum of frequencies: consider the noise of a pressure cooker. We also know about the unpleasant deprivation of the harmonics of a sound given by computers. What effects do they have on those who are listening? Is there a relationship between this aspect and the difficulties of an autistic subject? It seems that the topic has not been thoroughly discussed by the scientific community yet.

**Visual sensitivity.** As for visual sensations, it is scientifically proven that dazzling lights can cause problems to people with autism, as to people with epilepsy. The person with autism tends to have a strong preference for natural lights rather than artificial or intermittent lights. Animations, such as gifs or autoplayed videos, can distract from the main content. The problem for autistic people occurs when they cannot personally control their actions, therefore everything that is automatic, lacking lead, causes them discomfort. Another very common issue in people affected by autism is the preference/hate for some colors. Indeed, they prefer soft tones to bright colors, with a particular preference for green and blue [10]. A study conducted by the universities of Rennes and Kyoto over a sample consisting of 29 autistic children and 38 Typically Developing boys aged between 4 and 17 years has shown this strong predilection for green and blue, with the addition of brown. It has also shown a marked aversion to colors such as yellow, red and pink.

We want look for a moment at the Munsell color system: this will be useful to understand the studies on the relation between autism and colors perception. The Munsell system is a color space created by Albert Henry Munsell in the 20th century and used as an international standard to define colors based on three dimensional coordinates: hue, brightness and saturation. The system consists of three independent dimensions, each represented by a cylindrical coordinate system. The first coordinate is the hue, which is measured in degrees on a horizontal circle. Then there is saturation, measured radially from the neutral axis of the grays outwards. Finally we have brightness, which is measured on the gray axis from 0, which is black, to 10, which is white.

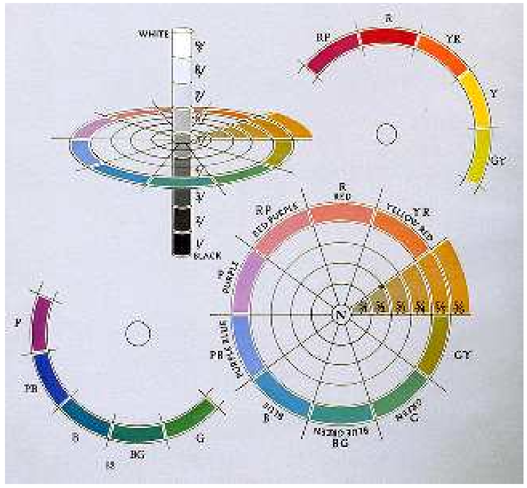


Figure 1.5: The Munsell Color System [36]

Munsell determined the positioning of colors in the system by measuring the human perceptual response to colors. The system schema is shown in Figure 1.5. He divided the circle into five main colors, red (R), yellow (Y), green (G), blue (B) and purple (P), then into five other intermediate colors adjacent to them. These shades are in turn divided into 10 sub-categories. The sum of two colors gives gray, while we can find two complementary colors on the opposite sides of the circumference. Saturation is measured radially from the center of each circular sector of the cylindrical coordinate system: it is the degree of color purity.

The colors employed in the study cited before have been proposed with the Munsell's system color-codings: red 7.5R, 4,14, yellow 10Y, 8.5,12, pink 7.5PR, 6, 10, green 2.5G, 3, 8, blue 10B, 7, 8 and brown 10R, 3, 10.
Repeated patterns can also cause them feelings of discomfort, as they alter

reality, destabilizing the person (see for example the captured video frame in Figure 1.7).

Therefore images should be simple, without few colors, they should represent concepts directly, without giving space to double meanings, they must be immediate and logical [22]. It has been proven that autistic people prefer photographs to other types of images, because they show reality [25]. An article published in 2015 shows that the greatest interest for an autistic is the center of the image and the objects in it that can be manipulated, rather than in the faces portrayed: they do not seem to be interested in the distribution of the objects or in the overall meaning of the picture (to get an example, see Figure 1.6) .

Many organizations have tried to reproduce how people affected by the Autism Spectrum Disorder see the world around them. In particular, from 2015 to October 2017, the National Autistic Society launched the Too Much Information campaign, which presented videos (see, e.g. Figure 1.7) and information to raise awareness on this topic [37]. The campaign focused on various issues. It dealt with sensory overload, drafting some videos on how autistic people perceive reality. It also discussed the work obstacles an autistic person might have: in fact, they start facing barriers during the work interviews, which may present many sensory distractions, and these barriers might be found also in communication and interactions with the colleagues, in the work organisation and in many other fields. The campaign also dealt with the problem that unexpected habit changes can create in autistic people with compulsive obsessions. The title Too Much Information is based

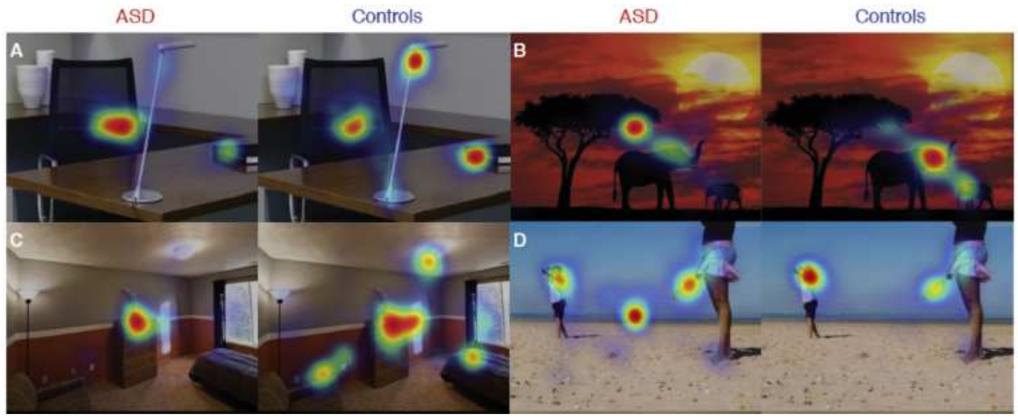


Figure 1.6: Autism images perceptions [25].

precisely on the idea of information overload that an autistic person is forced to face in different fields.

## 1.4 Therapies

Since this disorder is pretty unclear to medicine, some alternative techniques have been elaborated by pedagogues and operators to help autistic people, in particular non-verbal individuals. These techniques have been developed to make them express their feelings and thoughts differently from the classical ways.

Some examples are the Speech Therapy, which tries to improve verbal, non verbal, and social communication. Under this therapy we can find the Augmentative and Alternative Communication (AAC) techniques [17], which consists in a set of strategies based on the use of symbols, signs and pictures to help the autistic person to express himself and his thoughts. It is tailored

Figure 1.7: A frame from a video from the Too Much Information Campaign.

to a specific person, and can can also rely on the use of tablets and speech output devices: a pathologist identifies which is the best approach for each case. AAC can be aided or unaided. If it is aided the operator supports the individual with a computer, a tablet or a simple alphabet board or figure board (an example can bee seen in Figure 1.8). If instead it is unaided, the communication is based on gestures. One specific technique of AAC is the Facilitated Communication (FC), which makes use of the support of a *facilitator* who helps the disabled person forming words through a keyboard or an alphabet board. However, it is mandatory to say that FC approaches are not accepted by medicine, even if in some cases they seem to help for real, but FC is subject to an ethical debate because of the presence of the facilitator, which has a controversial role and should not express the real thoughts of the ASD-affected person or even take advantage from his disability.

People with autism can be also helped by behavioural treatments: the most famous are the *Applied Behaviour Analysis (ABA)*, the *Occupational Ther-*

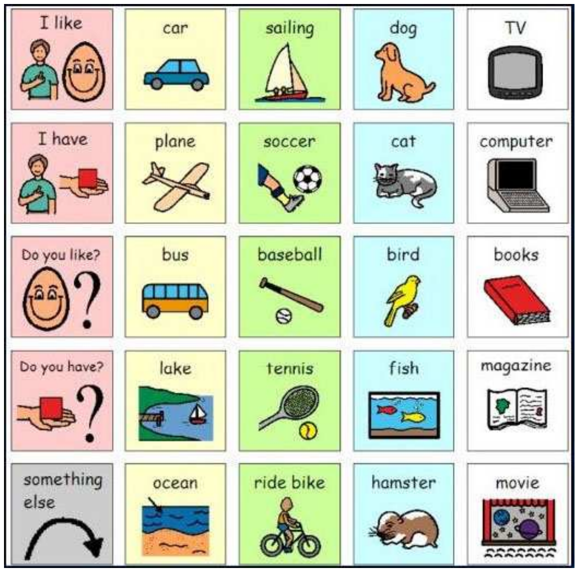


Figure 1.8: Example of communication board [38].

*apy (OT)*, the *Relationship Development Intervention (RDI)* and the *Treatment and Education of Autistic and Communication Handicapped Children (TEACCH)*.

The Applied Behaviour Analysis is based on positive reinforcement, and follows three steps: Antecedent, Behaviour, Consequence [7]. The first step is what occurs before a particular behaviour of the individual. The Behaviour is the person's response to an external stimulus and it has a target established by the operator. Then the Consequence is the reaction to a behaviour. The operator asks for something to the ASD person, then reacts to his behaviour by rewarding him in case of positive behaviour.

The Occupational Therapy focuses on well-being through occupation [12]. Occupation is any relevant activity for the individual, related with the age:

the autistic person becomes psychologically mature and learns how to relate with the others and to contribute to the community's life. Usually the activities are divided into self-care, work or school and free time. The therapist sets some goals to be reached and modifies them continuously. Following these indications the patient should increase his autonomy.

The aim of the Relationship Development Intervention is to increase relations and communication [1]. With RDI the parents are involved into their child's therapy, they become his trainers. Indeed, the treatment starts by restoring the meaningful relationships and evolves by gradually interacting with someone else. The idea is to guide the child during the activities: playing together will help to increase their relationship. When the child becomes more confident he can become part of a group of other children and learn how to relate with them.

One specific RDI program is the TEACCH program [40]. This program wants to support inclusion for people with disability within the family, at the workplace or, in a wider sense, in social life. During the sessions the children learn the concept of Space and Time: they understand where and when some actions should be done and the reasons why they do them. They also learn how to apply these activities to normal life, becoming more autonomous as the program proceeds.

Another possible treatment is *Music Therapy* [9]. It is mainly applied to children, but it can also be used with adults. It consists in giving a musical instrument to the autistic person: the instrument can be a tambourine, a bell or even an orchestra instrument. Being able to play instruments with their own hands allows them not to feel negative and to communicate through them. Music helps the autistic children to increase focus and attention, reduces anxiety and improves social behaviours. Under this therapy we can also consider the *TOMATIS Method* [41]: it has been developed by Alfred Tomatis, a french doctor and otolaryngologist. He identified the process of listening, as something which actively involves the ear: the idea is to stimulate the ear to increase the active listening process and so increase attention. The person is subjected to listening through an instrument called *TalksUp*, which passes from high to low frequencies causing a sudden adaptation by the ear and a stimulation to the brain. The listening sessions are divided into passive, in which the person merely hears the music with special headphones that transmit sound through air conduction and through a vibration on the bones of the skull, and active, in which the person reads aloud and listens his own voice through the headphones. Tomatis indicates Mozart's music as the most suitable for the passive listening session, due to the presence of variations in volume, tonality and rhythms [24].

The treatments that we have presented are only part of a big family of therapies that can be applied not only for Autism, but also for many different disabilities. Since medicine still have troubles to define all the peculiarities of ASD, we have a lack of scientific data on their effectiveness. For this reason these approaches shall be examined individually, depending on the situation and on the level of the disorder.

**Conclusion.** In this chapter we have seen in detail what is the Autism Spectrum Disorder, guiding the reader through the terminology that will be adopted also in the next chapters, and discussing on the data we have and on the therapies with which this disorder is treated. We have also seen the main difficulties an autistic person may encounter visiting a Web page. This chapter provides the basis for understanding the purpose of our work and the critical issues we met.

# Chapter 2

# Web Accessibility

In this chapter we will define Web usability and accessibility, exploring the protocols and guidelines given by the World Wide Web Consortium and the International Organization for Standardization (ISO). Then, we will present the Stanca Act and discuss the changes made from its first publication to its last update. Note that the accessibility guidelines we discuss here are very general and are meant to work with all types of disabilities.
The second section is dedicated to Web accessibility for people with Autism Spectrum Disorders.

## 2.1 Usability, Accessibility, and Inclusion

Usability is defined by the International Organization for Standardization (ISO) as follows:

**Definition 2.1.1.** [43]. ***Usability*** *is the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency,* *and satisfaction in a specified context of use.*

The previous definition gives us three parameters to evaluate usability of a Web content: the *effectiveness*, which is the precision and completeness of the Web content, the *efficiency*, which is given by the resources employed by the user to understand the content, and the *satisfaction* of the user by reaching the content. To give a simpler definition, a Web content is usable when the user who sees it for the first time does not need any help for reading and browsing the information. When we start to design a new website indeed, we must think about what we want to build and its final purpose and then we should imagine any possible difficulty the user may find while visiting the site content. The architecture of the website must be clear, the interactions with the user must be simple and the contents complete.
For these reasons in 1990 Jakob Nielsen developed some usability heuristics for user interface design, which became ten in their final set in 1994 [16]. The heuristics are divided into 3 categories and they can bee seen in Figure 2.1.

**Orientation and Navigation.** This category is about the orientation of the user inside the website: we should always let the user know the path he followed up to the page he is visiting, and where he can go next. Each section must be presented with a significative name, the user should understand everything without any limit and should be free to explore without getting lost.

**Error prevention and Management.** This category tell us that we should avoid accidental errors, but even if we cannot prevent all the possible bugs or exceptions, we should always give an explanation for the error and the possibility to come back to the previous state.

**Internal Consistency, Adherence to Standards, and Web Constraints.** This category contains the heuristics about consistency: the entire style of the Web content should be homogeneous and clear, with recognizable elements. Nielsen principals are not always shared by graphics designers: it seems like Nielsen is overshadowing graphics to support only simple and short contents. However, graphics has a key role in Web usability: even if it is supposed to be simple, it might be functional to the website communication.

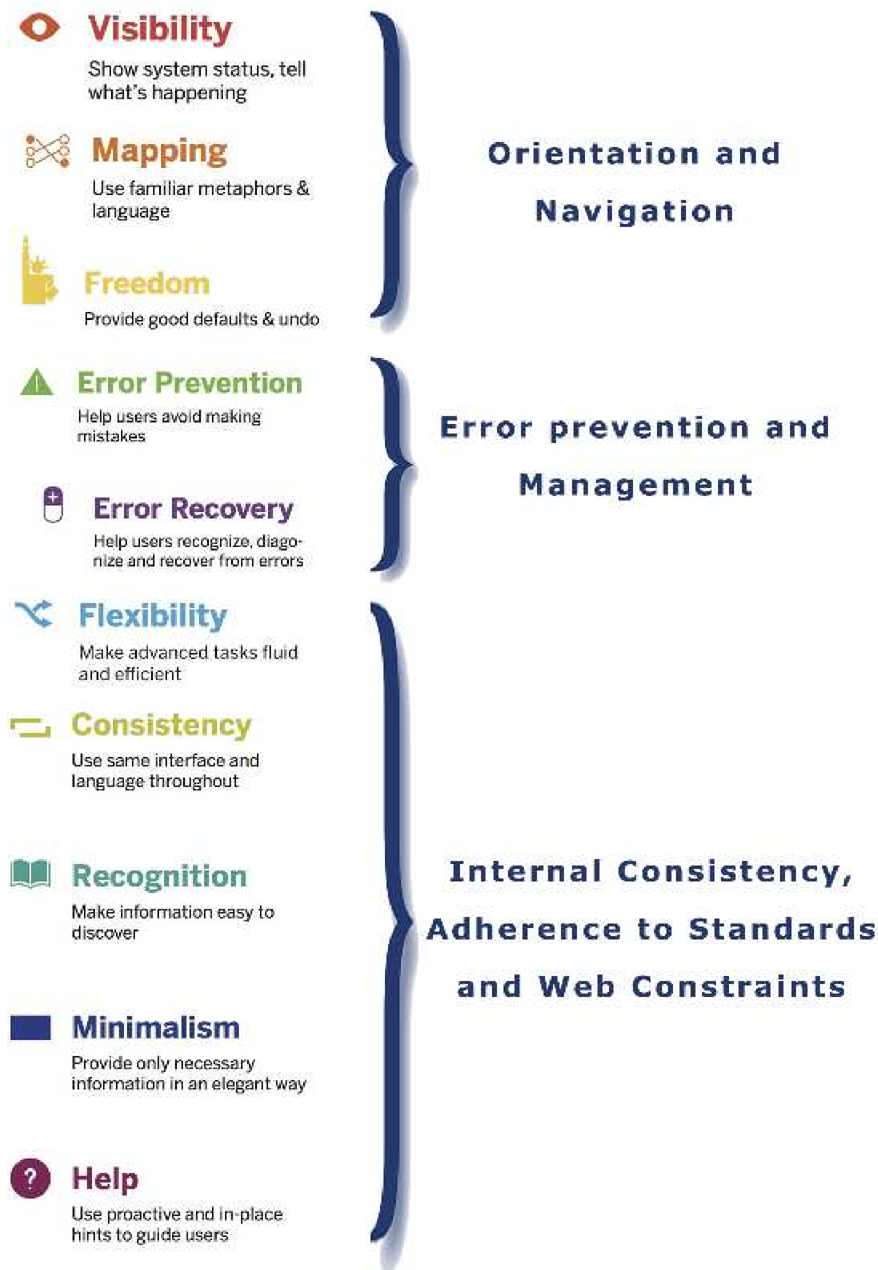


Figure 2.1: Nielsen Heuristics [16].

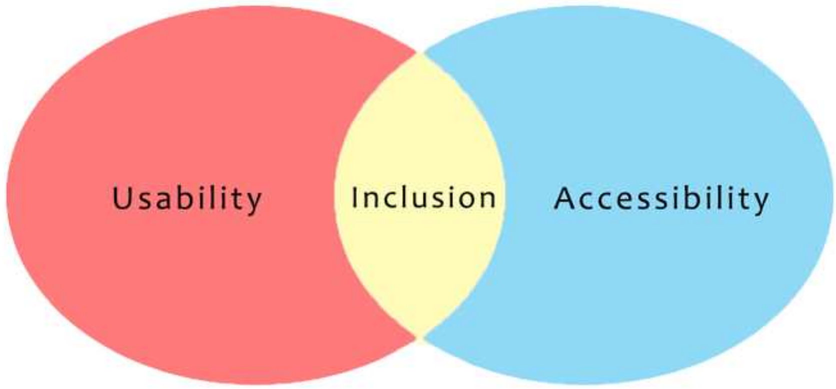


Figure 2.2: The relation between Usability, Accessibility and Inclusion.

While usability defines the easiness for a user to interact with the tool, *accessibility* is the usability of a tool for people with the widest range of capabilities. This concept is not limited to the Web, but can involve many fields where a service has to be offered, to make it available to all types of users. These two notions are strictly related with *inclusion*, whose purpose is to involve everyone to the maximum extent possible (see Figure 2.2).

Web accessibility stands for the possibility to access on-line information and services by any user and device, destroying any virtual barrier. It is primarily directed to people with disabilities, but it also involves all the other users, such as elderly, people using obsolete devices or technologies, etc. In the last years Web accessibility is grown thanks to many organizations and laws: in particular, the non-governmental organization **W3C-World Wide Web Consortium** introduced the Web Accessibility Initiative (WAI) in

1997 with the aim of make Web contents universally accessible [44]. This group guarantees accessibility for technologies and standards from the W3C, promotes research and training about WAI and defines documents addressed to three categories of users:

– Guidelines for accessibility to Web contents, which are for designers and authors;

– guidelines for accessibility of authoring tools, which are for the creators of authoring tools;

– guidelines for accessibility of Web navigation, which are for the developers.

Since technology changes and improves over the years, the principals described in the documents are generic and stable, and they are based on the following two ideas:

1. Multi-model content, which means pages that are still accessible despite physical, sensory and learning disabilities: it is good to have text and audio alternatives, however pages have to be accessible from any type of device and not only for a specific hardware, and so on.

2. Understandable and navigable content, which means simple language, evocative icons, etc.

The key idea is to merge these two concepts producing an equivalent content, that is something which can completely replace the official content giving the same informations: for example, images should be transformed into text, which can also be read with Braille or with speech synthesis.

### Web Content Accessibility Guidelines

The guidelines published by WAI are named **Web Content Accessibility Guidelines (WCAG)** and they are periodically updated. We have three WCAG versions, the first one is WCAG 1.0, published in 1999, the second one is WCAG 2.0, published in 2008 and the last one is WCAG 2.1, published in 2018.

For what concerns WCAG 1.0, each guideline is made by its identification number, its aim, its logic, the recipient categories, and a list of control points, which in turn are made by their number and their aim. They are organized by priority, from 1 to 3:

- Priority 1: Web developers *must* comply with this control point. This is a base requisite to allow some users using the website.

- Priority 2: Web developers *should* comply with this control point. This can help removing some virtual barriers for the access.

- Priority 3: Web developers *can* comply with this control point. This improves accessibility.

Given the number of priorities which are satisfied, WCAG 1.0 defines three conformity levels: A, AA and AAA. The Conformity Level A complies


Figure 2.3: The Four Universal Principals of Accessibility [39].

all the points with priority 1, the AA complies all the points with priority 1 and 2 and the AAA all the points with priority 1, 2 and 3.

Since WCAG 1.0 were related to a single technology, the updates that leads to WCAG 2.0 where necessary to make them stable. WCAG 2.0 are organised basing on the four universal principals of accessibility: perceivable, operable, understandable, robust (see Figure 2.3). A content is *perceivable* when information and user interface components are presented in perceivable ways for the user: for example, if the user is blind, he should access the content by audio components. A content is *operable* if the user can interact with many different input systems such as keyboards, mouse, pointers, etc.. It is *understandable* and *robust* if it can be easily interpreted by the user and by a variety of user agents.

These principals are guaranteed by 12 guidelines for WCAG 2.0 and 13 for WCAG 2.1. The conformity levels are redefined from WCAG 1.0, keeping the same names: A complies all the criteria for the previous conformity level A, or gives an alternative version which satisfy them; AA complies all the criteria for the previous A and AA or gives an alternative version which satisfy them, while AAA complies all the criteria for previous A, AA and AAA or gives an alternative version which satisfy them.

WCAG 2.1 are an expansion of WCAG 2.0 with some new criteria. The most relevant difference between the previous WCAG and the last one is that many criteria are now elaborated to be manually tested instead of automatically tested with some software tools: the reason is that technology might not identify some issues that a human check could find. These criteria has been elaborated specifically for people with learning difficulties and cognitive disabilities. Anyway, the structure is always the same both for WCAG 2.0 and WCAG 2.1. The WCAG 2.1 stucture is shown in Figure 2.4.

### 2.1.1 The Stanca Act

In Italy on 9 January 2004 has been emitted the Legge n.4, better known as the **Stanca Act** [34], from the name of the minister Lucio Stanca. This law gives specific guidelines about Web accessibility for people with disabilities. The directions provided by the law are compulsory for institutional and public websites, while they are only indicative for the companies' internal

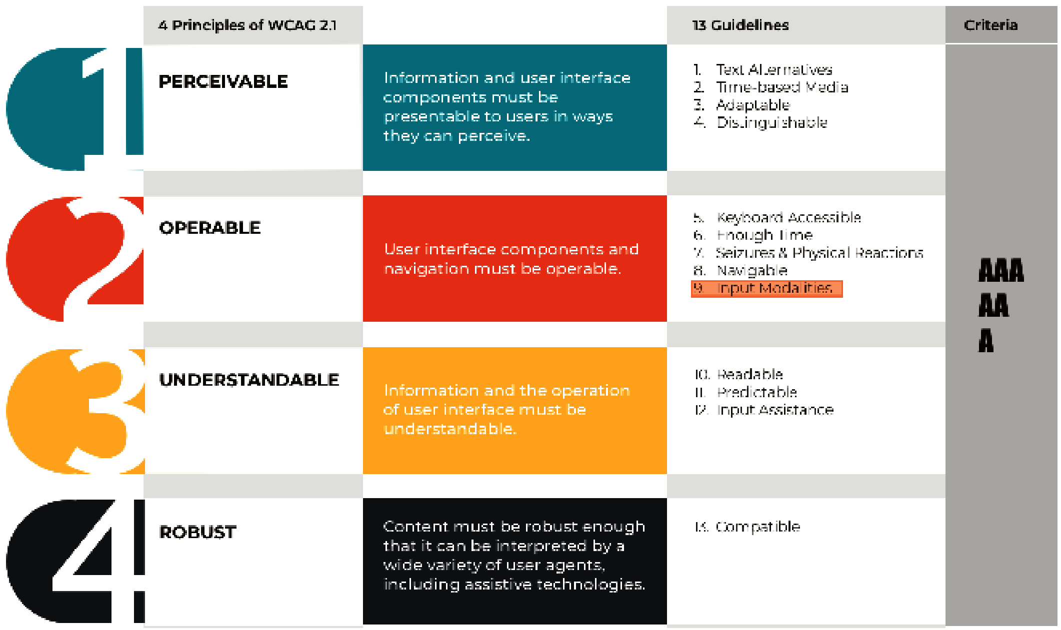


Figure 2.4: How WCAG 2.1 is structured. The new guideline has been highlighted.

websites, not addressed to the public.

The law has been changed to be aligned to the European standards for accessibility: in the first release of the law it had twelve articles, but it was updated on the last 11 September 2018 from a legislative decree of the 10 August 2018[1] with a redistribution of the articles and the addition of some sub-articles. In the previous version, the articles of the law presented its final purposes, the definitions of accessibility and assistive technologies and

[1] https://www.gazzettaufficiale.it/eli/id/2018/09/11/18G00133/sg

the recipients of the law. The law defines also the accessibility requirements, tools and accessibility guidelines.

The most relevant changes are in the first articles. The second article, which was about some definitions, has been extended with the definitions of mobile applications, website, extranet and intranet contents and assistance technologies: since these technologies were not known in 2004, they needed to be included in the new version of the law, in order to check accessibility for these new technologies.

In the third article, in addition to the websites providers to which the law refers, we can now find the four principals of accessibility seen above for WCAG 2.0 and 2.1, in addition to the *easiness and simplicity of use*, the *use efficiency* and the *use satisfaction* concepts. Indeed, in its new version, the law establishes that a Web content is accessible if it is usable, understandable, robust, perceivable, operable, easy to use, efficient.

## 2.2 ASD Accessibility Guidelines

In the previous section we have shown that there are many guidelines for making the Web accessible to people with different disabilities, but it is difficult to validate whether a Web content has been built specifically for people with autism: since this disorder does not present a specific set of symptoms such as blindness or deafness, but there is a wide range of limitations and there are many levels of disability, we can only generalize the instructions given by the W3C and try to provide new specific guidelines.

Let us now discuss how the problem of Web accessibility for autistic people is dealt. The WCAG previously mentioned has already given us some important recommendations. In particular, they provide some key criteria, which are the following:

- Audio Control: avoid making audio play automatically, leaving control to the user.
- Pause, Stop, Hide: objects subject to movement must have an option to be stopped, as well as content that updates itself, such as weather or time tools. There must also be an option allowing to check the frequency of the movement and of the updates.
- On Input: forms must not auto-send information when the fields are filled in. The focus of the user should not be changed by automatically jumping to the next field when the field is completed. Check boxes such as ”yes” or ”no” must not send anything before having clicked the ”send” button.
- Consistent Navigation: the navigation menus must always be in the same position in each page, always in the same order. This also applies to all other elements such as logos and particular icons.
- Consistent Identification: the icons must be consistent: if for example a printer is represented, the user will expect to be able to print by

clicking on it. Items with the same function must be labeled in the same way, the label must represent them based on their usefulness.

*Consistency.* This is another key concept for disability in general and for ASD in particular: each element should have its own role. We should be able to distinguish between buttons, links, simple text, images: it should be all uniform and recognizable. The common elements in user interfaces such as logos or navigation bars should always appear in the same place for each page of the site or application, the visitor should be able to find their way around and find them easily. This is a good approach for all types of users, but if it is useful to avoid slowdowns for highly functional users and keep their interest high, it is extremely important for people with ASD and their concentration.

*Web content.* When we visit a website or open a Web application we expect to find some simple lines with all the information we need. The idea of quick and simple information is the basis for Web accessibility, but it becomes essential for people with autism. When we build a Web content for ASD we should not create a wall of text, but it would be better to divide it into short sentences and use lists and bullets.

*Language.* The language adopted is another problem we must face: we must remember that TD people can understand abstraction, sarcasm, metaphors and figures of speech, while an autistic person cannot. For this reason, when we write a text for these people we should consider writing it in the simplest, most logical, most accurate way possible, although, as we shall see in the next chapter, we still have no way to automatically check the pragmatism of a text. Communication can also be helped by spacing out the text with simple images.

*Images.* Another aspect to be treated is the complexity of an image. It is even more difficult to be calculated than that of a written text. Optical illusions, which somehow distort reality and move away from logic, become particularly difficult to understand for people with ASD. However, they are not the only ones which have difficulty to understand ambiguous images, in fact this often happens even in the absence of disorders. This topic in particular has been studied in the Gestalt psychology, especially in many of the works by Professor Max Wertheimer. He was the first to study the rules that shape our visual system in the formation of perceptive units, developing the *Laws of Figural Unification* in the Twenties, which still constitute the basis of the most widely accepted explanation of our way of perceiving the outside world. The autistic person shows difficulties in amodal completion, in the distinction between the figure and the background, since the perceptions of distance and depth are deficient. Likewise with regard to moving images, since the speed of the objects nearer to us or of the larger figures is normally perceived as greater than that of the smaller and more distant ones. This is the parallax of movement, the resulting kinetic gradient which allows us to distinguish objects correctly but which generates a state of confusion in the autistic person who is not able to ”accommodate”. The sensory input starts the recognition process determined by the knowledge we already possess, but there are cases in which brain lesions at the neural level determine forms of visual *agnosia*[2] or *prosopagnosia* [3], which manifests itself the impossibility of recognizing faces [21]. The autistic person can find themselves in similar conditions if there is an overload of information, since they tend to focus on details and as such find it difficult to discriminate and discern the most relevant stimuli proposed to them: they struggle to inhibit their brain's automatic response, which leads to impulsiveness and nervousness.

The studies that have been developed so far focus mainly on the recognition of the objects represented, rather than on the ability of a person to understand their meaning. For example, we could consider the color distribution through a histogram of the image, or calculate the entropy through Shannon Information theory. However, these methods only give us the color distribution and the positioning of objects within the image. We could also try analysing the image with a recognizing approach, that is through an algorithm that could recognize it within a range of images given as input in the training phase. However, this could make the task harder because, as we will see in the next chapter, the complexity in understanding is something that depends on variables such as the knowledge and age of each individual. There are still no image databases built specifically to train an algorithm to recognize images and to label them with a complexity value that would

[2]Perceptual recognition disorder of objects, limited to a single sensory modality (visual, auditory, tactile, etc.)

[3]inability to recognize people's faces and, in the most serious cases, to distinguish one's own image in photography


Figure 2.5: ASD Accessibility Do and Don'ts.

allow us to understand whether an image is suitable or not for a person with autism. We do not even know if an image is more complex based on the number of layers it has or on the number of objects or colors it contains. In Figure 2.5 we can see a summary of the suggestions given to develop an accessible Web page.

As can be seen, these guidelines are too general, so they should be expanded and improved. We have already talked about the difficulty an autistic person faces when trying to stay focused: design in a Web content should be clean, simple, immediate. Unfortunately, many common websites have audios and videos which play automatically, auto-animated images or elaborate graphics (see for example Facebook or Giphy[4]).

Thus, since computer usage has been shown to have a positive impact on people with autism, guidelines have been revised and reorganize and new ones have been elaborated.

[4]https://giphy.com/

In this regard, in 2016 Dattolo and Luccio proposed detailed guidelines regarding the accessibility and usability of the Web for people with ASD [8]. These guidelines have been divided into categories [6].
The first one covers the graphical layout of a website, which includes everything that has to do with the appearance of the web content, therefore text, images, responsiveness.
The second one is about the quality of navigation and structure of a site, which reflects the aforementioned concept of consistency.
The third category focuses on the user's opportunities to customize the site in order to adapt it to their needs: for example, the possibility of using support structures such as vocal synthesis or color change, and more.
The last category is the one concerning language, with particular interest towards the complexity of the written text.
The guidelines have been summarized in Figure 2.6.

| | |
|---|---|
| G1 | The general design and the structure should be simple, clear and predictable, secondary content that distracts the user should be avoided. The number of features available at any time instant should be limited. |
| G2 | The content should be predictable and should provide feedbacks. |
| G3 | Pictures should be copiously used together with redundant representation of information. |
| G4 | Pictures can be drawings, photographs, symbolic images, should be easy to understand, should not go in the background, should be in a sharp focus. |
| G5 | Background sounds, moving text, blinking images and horizontal scrolling should be avoided. |
| G6 | The text should go with pictures. It should be clear, simple, and short (at most one sentence on a line); should be in a big font (14), in plain Sans-serif style (e.g., Verdana), in a mild color. Headings and titles should be used. |
| G7 | The interface should be responsive. |
| N1 | Navigation should be consistent and similar in every page/section. |
| N2 | The website and every mobile applications should have a simple and logical structure. Even the first time, the user should be able to easily navigate inside, and should remember the navigational information even at successive visits or uses. |
| N3 | Add navigation information and navigation buttons at the top and the bottom of the page. |
| N4 | Efficiency and availability. |
| U1 | Allow customisation. |
| U2 | Try to engage the user. |
| U3 | Make adaptive the interaction with users, considering their interaction history, their preferences, requests, and needs. |
| U4 | Decompose the tasks into simple subtasks. |
| U5 | The number of errors should be limited. |
| L1 | The language should be simple and precise. |
| L2 | Acronyms and abbreviations, non-literal text, and jargon should not be used. |

Figure 2.6: Accessibility and Usability Guidelines for users with ASD [6].

The features described above can be identified with the letters G1-G7 for the first category, N1-N4 for the second, U1-U5 for the third and L1-L2 for the fourth. As we can see, many of these guidelines, although already existing, have been extended and adapted to the specific disability.
Let us start with the ones in the first category. While G1 and G2 were already mentioned in the general guidelines, starting from G3 we see new suggestions. Here, in fact, it is suggested to use many that images in or-

der to explain the content of the written text, and that information must be specified more than once. This means that it is more correct to provide information in several different ways, in order to allow the user to process it, without, however, weighing it down, so as to keep interest and concentration alive. Again, in G4 they refer to the type of images that can be used, with particular attention to the complexity of their comprehension and to their positioning: an image in the foreground, well highlighted, is preferable to an image positioned against the background of other contents. With G5, sounds, text or moving and disappearing images are banned. In G6 it is suggested to lighten the text, to make it immediate with the use of a highly readable Sans-serif font, in a mild color, interspersed with images and divided into sections. In G7 it is suggested to create responsive interfaces, namely adaptable interfaces based on the size and resolution of the screen of the device from which the site is consulted.

The second category, already present in the previous guidelines, is shared here: people with autism spectrum disorder need linear and logical structures, which do not make them lose their orientation.

The third category adds some interesting suggestions for usability. U1 talks about customization, or the possibility for the user to modify what he sees based on his disability. This implies being able to change colors or to “clean” the page by eliminating all what is not necessary and which is a source of distraction. Another issue is user engagement: it is important to allow those who access the site to find what they are looking for as quickly as possible, the user must believe that the site has been designed exactly for what he

needs. All infrastructures must therefore work well and quickly, the contents must be relevant, reliable and easy to find.
The last guidelines are related to language, we will see in the following chapter how this topic is addressed.
A recent survey conducted by Yu, Murrietta et al. of the Southern Methodist University [28] evaluated six hundred websites designed for people with autism dividing them into four main categories and ranking them based on friendliness: websites focused on Autism and based on the studies of Dattolo and Luccio [6], federal websites that meet the minimum regulation required for disabled users, websites that appear on Google Search when using keywords like "autism friendly websites", and websites with the most intense Web traffic according to Alexa [29]. This article is based on previous studies on the usability of websites for people with autism conducted by Nikolay Pavlov, at the Plovdiv University of Bulgaria [18]. Each site was given a score based on how suitable it is for people with ASD. The final rank was obtained by dividing the different scores into 5 tiers, from the most autism-friendly, in the lower tier, to the least friendly, in the higher tier. This score is based on the number of the features of the site that meet the needs of a person with autism. In many cases the websites even if have fewer animation and may be classified as autism-friendly, present an insufficient variety of font sizes and typology, so they are classified as autism-unfriendly.

**Conclusion.** This chapter covered the issue of Web Accessibility for people with disabilities. In Italy, Europe and all around the world the governments and the organizations are trying to propose universal rules to make the World Wide Web accessible for anyone; many countries are trying to elaborate a strategy to make usability and accessibility universally recognised and applied. We have also seen here how the ASD problems visiting a Website are addressed from the International Standards Organizations and the W3C and by the research papers.

# Chapter 3

# Natural Language Processing

In the previous chapters we have mentioned the importance of presenting simple and effective texts to people with autism spectrum disorder. Let us try to understand now how we can determine the complexity of a text, underlining some differences between the English and the Italian languages.

## 3.1 Languages and grammars

At the beginning of the 19th century, proposals were put forward by various scholars to classify languages into ”morphological types”. The languages spoken today are around 6000, some of which are on the brink of extinction due to the small number of speakers for both historical and political reasons that is not up to us to analyze. Here, we will only consider how, through the use of languages which are understandable by machines, it is possible to facilitate communication in particular for people suffering from autism spectrum disorders, but also for all those who may encounter difficulties in using these systems. At the turn of the Nineteenth century, F. De Saussure, a distinguished Swiss semiologist, laid the foundations to modern linguistics, in particular regarding the structure of a language. He introduced the concepts of langue and parole, where langue refers to the social aspect of a language, a set of shared signifiers and significances, while parole represents the individual aspect of a language, that is the personal linguistic outputs that can assume different values particularly with the poetic language. He therefore distinguished the concept of signifier and significance: the former is in fact the sound of a word, the set of phonemes and morphemes that determines it, while the latter is the abstract concept, the one that is meant for communication.

As we know, many important computer problems are based on string processing and search and this also applies to other areas of science. Here we will present some definitions which will be useful later to understand Natural Language Processing.

**Definition 3.1.1.** [23]. *An **alphabet** is a non-empty finite set of symbols (characters)* $\Sigma = s_1, s_2, .., s_n$.

**Definition 3.1.2.** [23]. *Given an alphabet* $\Sigma$ *we define **string** or word a finite sequence of characters of* $\Sigma$*. Given a string* $x$ *the number of characters that constitute it is called string length and is indicated as follows:* $|x|$*. The zero-length string (ie not consisting of any symbol) is called an empty string* *or nothing. The set of all the strings defined on the alphabet $\Sigma$ (including the empty string) is denoted $\Sigma*$.*

**Definition 3.1.3.** [23]. *Given an alphabet $\Sigma$ and $\Sigma*$ set of all the strings defined by the alphabet, a* ***language*** *$L$ is a subset of $\Sigma*$ formed by some of the strings that can be generated starting from the alphabet $\Sigma$.*

A language is therefore a set of symbol sequences, without any reference to what such sequences can mean.
Each language is governed by a grammar.
The Grammar constitutes the symbols of any type of language, establishing the rules for the construction of correct and comprehensible expressions. We can therefore provide the definition of a language L, which is the following:

**Definition 3.1.4.** [23]. *A* ***grammar*** *$G$ is defined by four finite and decidable sets $G = (NT, T, S, P)$ where:*

- *$NT$ is the set of non-terminal symbols*
- *$T$ is the set of terminal symbols*
- *$S$ is the initial symbol*
- *$P$ is the set of productions*

*Starting from the initial symbol, the application of the productions generates the language. The productions are the rules of the grammar. The language* *generated by the grammar is the set of terminal strings that can be generated starting from the initial token $S$.*

A necessary condition is that the intersection between the set of non-terminal symbols and of terminal symbols is empty: without this condition we would have metasymbols inside the strings of the language we are describing. Given a language on an alphabet, the first problem that is faced is its representation. In the case it is formed by a finite number of strings, the problem is resolved by simply listing them. However, languages can be formed by a non limited number of strings, so there exists different representation approaches: we will focus here on the *generative approach* and on the *recognition approach*, which are the most important.

**Generative approach.** The generative approach defines through a grammar the structural rules that must be satisfied by its strings. We have a finite number of symbols and rules and with it we generate an infinite set of sentences. The rules are specific impositions on the way of composing the sentences themselves. Thus, for example, if we have the letters of the alphabet as symbols and the English grammar's rules, we know that we can produce innumerable sentences with a complete meaning as "I eat an apple", but we cannot produce sentences without any sense such as "An eat apple I". In the generative approach, therefore, a language is defined as the set of terminal strings obtainable by applying a finite sequence of rewriting steps consisting in the application of output rules.

**Recognition approach.** Finally the recognition approach, provides a recognition algorithm that, received a string as input, tells if it belongs or not to the language. However, not all languages can be recognized by algorithms and not all languages can be defined through grammars. This type of approach exploits the so-called automata, which are abstract computers represented through states and transitions. They take a finite set $I_n$ as input and output a finite set $O_m$, produced through an algorithm. Automata are divided into categories, which are, just to mention them, the *Turing Machines*, the *Limited Linear Automata*, the *Stacked Automata* and the *Finite State Automata*.

Grammars were classified in 1957 by Noham Chomsky [3], an American linguist whose research also covers the psychological, logical and philosophical fields as well as supporting political writings of a radical yet pacifist orientation. For Chomsky syntax is the core of any language. He claimed that only in the human mind does a language processor exist, which would explain the incredible speed with which a child acquires a language. Chomsky believes that there is a common syntax for all languages that is adapted with few parameters to become a concrete language. In computer science, he has helped to create formal languages that are the basis of modern programming languages. Chomsky classified the grammars according to the type of production they use, categorizing them into type 0 grammars, type 1 grammars, type 2 grammars and type 3 grammars.
Grammars are classified in decreasing order of complexity: starting from the

more general type 0 grammar, and gradually adding constraints, we arrive to type 3 grammar. Type 0 grammars are not limited ones, based on more general productions. These grammars produce any type of language, without constraints, and include all the others. The production rules have no restrictions. Languages produced with a type 0 grammar are also known as recursively enumerable languages or type 0 languages. In the recognition approach, they have an expressive capacity equivalent to that of the Turing machine.

Type 1 grammars are based on contextual productions, and for this reason they are also called context sensitive. Compared to those of type 0 they introduce a constraint that limits the number of variables that are in the left of a rule. They generate type 1 languages. In the recognition approach they correspond to linear automata.

Type 2 grammars are based on non-contextual productions, so they are also called context free, and generate type 2 languages. Compared to those of type 1, they introduce the constraint that the left side of a rule be made of a non-terminal symbol. This constraint prevents the presence of a context that could influence the activation of a rule. In the recognition approach they correspond to the Stacked Automata.

Type 3 grammars are those that rely on regular productions, and generate type 3 languages. Compared to type 2 grammars, the constraint that the right part of a rule be made of a terminal symbol is introduced. In the recognition approach they correspond to finite state automata.

A diagram of this organization of the grammars is given in the Figure 3.1.

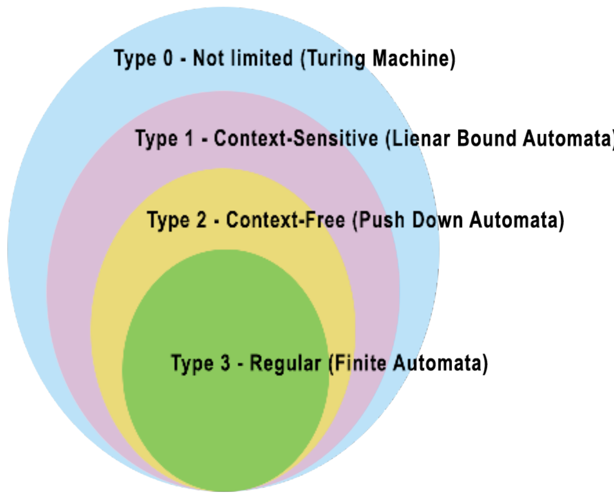


Figure 3.1: Chomsky Hierarchy [3].

A language generated by a grammar can also be generated by another grammar, equivalent to the first. Two grammars are called equivalent when they generate the same language.

## 3.2 Natural Language versus Formal Language

After this long digression, we finally have the basics to define what is meant by the term *natural language* and describe its opposition to the *formal language*. By *natural language* we mean a language developed and affirmed spontaneously in the human cultures. In fact, the natural language is the idiom of a nation. Each social group shares its own natural language, with its expressive richness and its nuances. The problem is that the natural language is in itself an ambiguous language, which cannot be easily formalized. Thus, from a mathematical point of view, it is opposed to the formal language described above, while from the IT point of view its opposition is found in programming languages, used to write instructions to be given to a computer. Establishing a formal language allows us to understand if a sentence is legitimate, if it belongs to a particular language.

The formal language as the natural language has a syntax, too, which is the set of rules necessary to build correct sentences for that language. It is expressed through formal notations. The formal language also presents a semantics, which is the set of meanings to be attributed to the sentences. A sentence can be syntactically correct, but not meaningful. The semantics can be expressed through words, in a not very precise and ambiguous way, through actions, mathematical functions or logical formulas.

The formal language as the natural language also has a syntax, which is the set of rules necessary to construct correct sentences for that language. It is expressed through formal notations. The formal language also presents a semantics, which is the set of meanings to be attributed to the sentences. A sentence can be syntactically correct, but not meaningful. The semantics can be expressed only in words, in a not very precise and in an ambiguous way, through actions, not mathematical functions or logical formulas. A language is evaluated by an interpreter, who takes the individual sentences in input by executing them one at a time. The language is rewritten by the compiler, which accepts in input an entire program written in L and rewrites it in a simpler language.

It is possible to produce a lexical analysis of a sentence, which consists of identifying single words (tokens) of the sentence classified in appropriate categories. The categories can be names, keyword and punctuation marks, among others. The purpose of the lexical analysis is to divide the text into lexical units, i.e. groups of fonts called lexemes. The lexemes are then classified according to their role, called token class. Token classes are sets of strings. The result of the lexical analysis is therefore a sequence of $< tokenclass, string >$ pairs called tokens.
In a sentence one can also start the syntactic analysis, which consists in verifying that the sentence, understood as a sequence of tokens, respects the grammatical rules of the language.
Semantic analysis determines the meaning of a sentence. A concept is associated with a sentence. It is therefore necessary to establish a function that associates each sentence with an element of some domain. The function must give meaning to each symbol, then to each word and finally to each sentence.

All this formalization of the language is determined by the objectives of the artificial intelligence, including that of putting an automaton in a position to express itself through natural language. In order to do this it is necessary that the programs understand and produce a speech in natural language. An example of artificial intelligence that takes advantage of these concepts is NETtalk. This program learns and pronounces an English text taken as input, comparing it to the phonetic transcriptions of the symbols. The writing and study of this type of programs is called natural language processing

(NLP). It is based on the concept described above of language generating grammar. Precisely to get around the problem of language ambiguity we use lexical, grammatical, syntactic and semantic analysis to process it.

## 3.3 NLP, NLP indexes and considerations

Given the necessary premises, we can finally talk about the processing of natural language, also called NLP. It is the process of automatic processing written or spoken natural language information by an electronic computer. This is a particularly difficult and complex process due to the intrinsic characteristics of the ambiguity of human language.
For this reason Chomsky and other scholars have tried to formalize the concepts of language and grammar, in order to make them understandable even for a machine. In fact, a machine is not able to distinguish the shades of a language, it cannot interpret them unless there are specific mathematical constraints that describe them. Understanding natural language is often considered an AI-complete problem, i.e. a problem whose resolution is considered equivalent to creating a realistic artificial intelligence [20]: this happens precisely because the comprehension of the texts requires the understanding of the concepts associated with them. To associate innumerable signifiers with a completely objective significance is therefore an extremely complex task.
In 1966 Eliza was developed, a program for human-machine interaction that simulates a psychotherapist. The sentences inserted by the user must be

completely arbitrary, and the system responds with another sentence, giving the impression that a real dialogue is taking place. However, this cannot be considered as the first natural language comprehension program, as the program does not understand anything of what is said, but only searches for sentences in a set of known patterns, printing a generic answer when it finds no matches.

### 3.3.1 Natural Languages Comprehension Tools

The first natural language comprehension program was developed at MIT by Terry Winograd between 1968 and 1970 and is called SHRDLU. The name derives from the sequence ETAOIN SHRDLU, which represents the layout of the keys on a Linotype machine, based on the decreasing incidence in the English language. SHDRLU allows an interaction based on the use of English terms. Through a screen the user is shown a virtual environment consisting of a flat surface, a box and a series of colored cuboid or pyramidal objects. The user can interact with an imaginary robot arm to move objects by simply communicating it to the program in English. SHDRLU is very powerful, in fact it is able to distinguish an object even when it is implied. To do this, Eliza submits what the user requires to a syntactic, semantic and deductive analysis [27].
A simple example of operation is the following:

USER: Take the blue block.

COMPUTER: Ok

USER: No, that is a pyramid.

COMPUTER: I don't understand what pyramid you're referring to

USER: Find a block that is higher than the one you are holding and put it in the box.

COMPUTER: With "put it" I take the block that is higher than the one I'm holding

USER: Yes

COMPUTER: Ok

USER: Put the block you are holding in the box

COMPUTER: Ok

As can be seen, the results are exceptional and this led to an excessive optimism on the part of the programmers, which diminished however as soon as the systems were extended to more realistic and complex situations. Despite this, the new vocal synthesis technologies, in which the human voice is artificially reproduced, are also based on natural language processing. It is no coincidence that these systems find difficulties in the presence of homographs, numbers or abbreviations, which must be translated with a correct phonetic representation: for example a four-digit number can be a pin, and therefore it must be read digit by digit, or a date, and therefore it must be read as such. The only way to understand this is to do a semantic analysis and derive it from the context, but as we have seen this is still a problem. Another problem arises when we face the pragmatic level of a language: when two humans talk, they exchange messages between them that can be interpreted as simple statements but also as implicit or other requests. To understand a sentence then it is not enough to take into account the meaning of the words, but also the situation, the context and the tone with which the sentence is expressed must be considered.

### 3.3.2 Semantics Analysis

Let us see how the syntactic, the semantic and the pragmatic analysis of a text is dealt with in the automatic processing.

We said that the syntactic analysis is the simplest: we analyze the sentence recognizing its grammatical correctness. The semantic analysis is much more complicated and in the simplest cases we proceed analyzing the meaning of the single words, then we obtain the syntactic structure of the sentence, and finally a possible meaning for the whole sentence. The problem is that a word can have different meanings (especially in some languages, such as Greek, a single word can take up many meanings). Furthermore, a different meaning can be attributed to the same succession of words, which can be figurative: think for example about the popular sayings or certain idiomatic phrases. Artificial intelligence tries to overcome this problem by relying on a knowledge base called training set, in which the machine learns a set of expressions in order to recognize them in the testing phase, but this implies having extremely long learning phases, which doesn't happen for a human being.

The analysis at the pragmatic level is the most complex: in fact it must also look at the intentions of the interlocutors, who are only partially reflected in their words. This cannot be formalized in mathematical terms, because it does not reflect any rule, except those of cultural nature in a broad sense established within a given human group and therefore extremely variable. Some attempts at pragmatic analysis have been made by the philosophers of the language, with the theory of the Speech Acts. It is based on the assumption that with a statement it is not only possible to describe its content or support its truthfulness, but that most of the statements serve to carry out real actions in the communicative field, to exercise a particular influence on the surrounding world. In simpler terms, to understand a sentence it is necessary to understand what kind of action the speaker makes in delivering their message, where by actions we mean affirmation, request, promise, etc. The listener must therefore create a series of mental states, discarding the least plausible ones until reaching the most plausible ones, thus finding the pragmatic meaning of the sentence.[13]

The NLP could be used to process huge sets of documents available online for the most diverse applications. In fact, there exists also the Statistical NLP, a set of quantitative approaches to natural language processing. It is based on information retrieval and their extraction and automatic translation. However, the vastness of the present documents has changed the approach to the study of word processing. Limiting ourselves to the analysis of the Italian and English languages, it is important to underline that the latter is very difficult to typify because, despite its fortune, it contains many inconsistencies and typological contradictions. However, it is precisely this feature that induces English to consider individual headwords within its own semantic field, which determines a better possibility of a comprehensive comprehension of sentences.[15]

### 3.3.3 Text Readability

At this point we must introduce the concept of readability, which differs in Italian and in English. In the former, readability does not have different words to distinguish between calligraphy and reading fluency in relation to the linguistic structure: the same text can be readable but lack an adequate linguistic structure. In contrast, in the latter these two concepts are summarized with the terms legibility and readability.

The readability of a text therefore depends on its content and on its presentation.

The readability exists for both natural languages, artificial languages and programming languages. We will neglect all the other languages, and focus on Italian and English. As we all know a language changes over the years,

with the introduction of new words and expressions and the abolition of the less used ones. This leads to a constant evolution of it, not only with regard to our vocabularies, but also to the composition of sentences and phrases. In the second half of the Nineteenth century Professor L. A. Sherman discovered that the sentences in English were halving in length compared to the times of Queen Elizabeth I. He also established that written texts are more complex to understand than speech, as they are deprived of facial expressions and tones that can instead be captured in conversation. The fact that the sentences were shortening, led him to elaborate the thesis that text was becoming more and more similar to speech, simplifying itself.
In 1889 in Russia Nikolai A. Rubakin published a study in which he stated that the main obstacles to understanding a text are unfamiliar words and long sentences. In the 1900s formulas were then elaborated to calculate how much a text is readable, in proportion to age and level of education of the reader. The level of education goes from elementary school, which ends with the 5th grade, to college (beyond the 12th grade, last year of high school). These indexes are based on the aforementioned studies, they calculate the average length of a sentence, the number of difficult or uncommon words, the number of personal pronouns, the percentage of words that appear only once and the number of prepositions.

### 3.3.4 Flesch Reading ease e Flesch-Kincaid grade level

In 1943 Flesch developed a readability formula to calculate the difficulty of a text with respect to people between 10 and the adult age. The *Flesch Reading ease* formula is divided into two parts. One part calculates the complexity of the text, the other predicts involvement based on the number of references and personal sentences. It uses a scale from 0 to 100, where 0 corresponds to the 12th degree and 100 to the 4th degree. The higher the result, the higher the comprehension of the text.

The following equations presents some numbers that Flesch and Kincaid deduced from their personal studies, by trial and errors: anyway, both of them are based on the two key factors of *sentence length* and *word length*, in fact the analysis of sentences that contain many long words gives an higher complexity than shorter sentences with shorter words.

Formula reading ease:

$$206.835 - 1.015 \times \left(\frac{total \quad words}{total \quad sentences}\right) - 84.6 \times \left(\frac{total \quad syllables}{total \quad words}\right) \quad (3.1)$$

The results can be interpreted as follows:

| Score | School Level | Notes |
|---|---|---|
| 100.00-90.00 | 5th grade | Very easy to read. Easily understood by an average 11-year-old student. |
| 90.0–80.0 | 6th grade | Easy to read. Conversational English for consumers. |
| 80.0–70.0 | 7th grade | Fairly easy to read. |
| 70.0–60.0 | 8th and 9th grade | Plain English. Easily understood by 13- to 15-year-old students. |
| 60.0–50.0 | 10th to 12th grade | Fairly difficult to read. |
| 50.0–30.0 | College | Difficult to read. |
| 30.0–0.0 | College graduate | Very difficult to read. Best understood by university graduates. |

Table 3.1: Flesch reading score interpretation.

The Flesch-Kincaid level of education has instead been developed for professors, parents, booksellers and all those who must judge the complexity of a text addressed to students. The level of eduction is calculated with the following formula:

$$0.39 \times (\frac{total \quad words}{total \quad sentences}) + 11.8 \times (\frac{total \quad syllables}{total \quad words}) - 15.59 \quad (3.2)$$

The result is a number that corresponds to the level of education for which the text was conceived. However, as the formula is constructed, the score does not have a maximum limit.

### 3.3.5 Dale-Chall formula

In Ohio, Professor Edgar Dale challenged previous studies that elaborated lists based on occurrence [5]. In fact, he decided to expand those lists, because they contained only one meaning for each word, not considering that words can have more than one meaning, as already stated. The *Dale-Chall* formula calculates the percentage of complexity of a text, which tells us to what extent the reader well be able to understand. This formula is interesting because unlike the others, that use word-length to assess word difficulty, it uses a count of *hard* words, basing on some dictionaries.

$$0.1579 \times (\frac{number \quad of \quad difficult \quad words}{number \quad of \quad words \times 100}) + 0.0496 \times (\frac{number \quad of \quad words}{number \quad of \quad sentences}) \quad (3.3)$$

The "difficult words" indicated in the formula are all the words that are not in the list of words provided, but we must consider that the list is made of basic forms such as names and verbs, so for instance the plurals and the verbs, among others, have to be added.

The results can be interpreted as shown in the table:

| Score | Notes |
|---|---|
| 4.9 or lower | easily understood by an average 4th-grade student or lower |
| 5.0–5.9 | easily understood by an average 5th or 6th-grade student |
| 6.0–6.9 | easily understood by an average 7th or 8th-grade student |
| 7.0–7.9 | easily understood by an average 9th or 10th-grade student |
| 8.0–8.9 | easily understood by an average 11th or 12th-grade student |
| 9.0–9.9 | easily understood by an average 13th to 15th-grade (college) student |

Table 3.2: Dale-Chall reading score interpretation.

### 3.3.6 Coleman-Liau index

The *Coleman-Liau readability index* was created in 1975 by Meri Coleman and T.L. Liau [4]. This index does not use the syllable count, but counts the word length in number of letters. It is widely used in the medical, legal and translation sectors. Coleman and Liau argued that syllable-counting techniques lacked accuracy and that it is also time-consuming. According to them, the number of letters of a word is a better predictor of readability.

The formula to compute it is:

$$CLI = 0.0588L - 0.296S - 15.8 \tag{3.4}$$

where $L$ is the average number of letters per 100 words and $S$ is the average number of sentences per 100 words.

This also gives us the degree of education necessary to understand a text.

### 3.3.7 Automated Readability Index

The *automatic readability index* has been developed for the English language [0]. This index also gives an approximation of the level of education necessary for the comprehension of a written text.

The formula is as follows:

$$4.71 \times (\frac{number \quad of \quad characters}{number \quad of \quad words}) + 0.5 \times (\frac{number \quad of \quad words}{number \quad of \quad sentences}) - 21.43 \quad (3.5)$$

where by fonts we mean the number of letters and numbers, by words the number of spaces and by sentences their own number. Non-integer scores are rounded up. Like the Coleman-Liau index, ARI is also based on word length based on the number of fonts that make it up. This makes the calculation of the lengths easier for the machines. Indeed, this index has been developed for real-time monitoring of readability on electrical machines. The correspondence of the score obtained, age and level of education can be seen in the following table:

| Score | Age | Grade Level |
|---|---|---|
| 1 | 5-6 | Kindergarten |
| 2 | 6-7 | First/Second Grade |
| 3 | 7-9 | Third Grade |
| 4 | 9-10 | Fourth Grade |
| 5 | 10-11 | Fifth Grade |
| 6 | 11-12 | Sixth Grade |
| 7 | 12-13 | Seventh Grade |
| 8 | 13-14 | Eighth Grade |
| 9 | 14-15 | Ninth Grade |
| 10 | 15-16 | Tenth Grade |
| 11 | 16-17 | Eleventh Grade |
| 12 | 17-18 | Twelfth grade |
| 13 | 18-24 | College student |
| 14 | 24+ | Professor |

Table 3.3: ARI score interpretation.

### 3.3.8 FORCAST

In 1973 an American military study developed a formula to understand the level of reading skills required for different military tasks. Unlike the other formulas, it analyses incomplete sentences through the use of a dictionary, it does not deal with sentence length, so can be used on texts without sentences, such as forms, surveys, questionnaires, multiple choice questions or texts containing lists or bullet points.

The *FORCAST* formula is:

$$20 - \left(\frac{number \quad of \quad single{-}syllable \quad words \quad in \quad a \quad sample \quad of \quad 150 \quad words}{10}\right) \quad (3.6)$$

### 3.3.9 Gulpease

The *Gulpease index* was defined in 1988 at the University of Rome and is an index calibrated on the Italian language [14]. As we already know, in fact, the Italian language is very different from the English one, both in terms of the structure of a sentence, and in terms of syllable counting, writing and word reading. Like the Coleman-Liau and ARI indices it uses the length of the words in letters instead of syllables. The index considers two linguistic variables: the length of the word and the length of the sentence with respect to the number of letters.

The formula is as follows:

$$89 + \left(\frac{300*(number \quad of \quad sentences) - 10*(number \quad of \quad letters)}{number \quad of \quad words}\right) \quad (3.7)$$

The results are between 0 and 100, where 0 indicates the lowest degree of readability while 100 stands for the highest. In general, texts with an index lower than 80 are difficult to be read by those with an elementary license, those below 60 are difficult to be read by those with a middle school certificate, while those under 40 are difficult to be read by those with a high school diploma.

Obviously all the studies produced in this field take into consideration only texts of argumentative or formal nature. Poetic texts totally escape from any classification of this type given the complexity of the interpretation of each word that is full of meanings to be deciphered.

**Conclusion.** In this chapter we have given various definitions about languages and grammars. As we have seen, it is difficult for a machine to be able to deduce the complexity of a text but on the basis of mathematical calculations, which however cannot grasp the shades of a language. Although artificial intelligence is very advanced, it is still not possible to define a unique method to understand how difficult a text is in terms of understanding. The substantial differences between languages make the problem even more complicated. However in the era of domotics and artificial intelligence we will try to expand our knowledge on the subject, to find new solutions and techniques applicable to learning machines and to encourage interaction with the user. Let us think, for example, of the voice assistants of our mobile phones and now also of our household appliances: Natural Language Processing allows to improve their vocabulary and their ability to understand the commands we give them.

What we learned here will be used in our validator when it comes to outlining the complexity of a text within a Web page. Here we only want to anticipate that for the English language will use the Flesh Reading Ease, since it is the most famous and, at the same time, it produces a result which is the most simple to understand. For the Italian language we will use the Gulpease index, since it is the only one among those we have discussed to be strictly dedicated to the Italian language, and therefore it is more precise for it.

# Chapter 4

# The Online Tools

This chapter is dedicated to the presentation of some online accessibility tools which are very general, but not directly dedicated to people with ASD.

## 4.1 Online Accessibility Tools

Wandering around the web one can find many tools for checking the accessibility of websites, texts and images. Although none of these tools were specifically designed for people with autism, they are certainly useful as a basis for understanding how a validator of Web pages for autism should be structured.

Here, we will deal only with a few of them which have a free or open source license, but it is important to remember that there are many more professional ones created for companies and other private entities. The following were found in the list provided by the World Wide Web Consortium[1].

### 4.1.1 508 Checker

The first validator proposed is 508 Checker[2] (see Figure 4.1). 508 Checker was made by the company Formstack, which deals with the collection of information through surveys, event registrations, online forms. It supports CSS, HTML, XHTML and automatically checks individual Web pages by inserting the url in the appropriate field. It generates a report in HTML format, checking if the page at the address provided complies with the guidelines of section 508, which is a federal law of the U.S.A which requires the government to use exclusively electronic and computer technology accessible to users with disabilities. The license is free, the release date is January 2014.

### 4.1.2 Access Assistant Community Edition

This free license tool[3] was developed in October 2017 by Level Access. The company has a team of legal, risk, compliance and technical experts, many of whom are themselves disabled. This makes them particularly sensitive to the issue of accessibility and for this reason the company's goal is to make

[1] https://www.w3.org/WAI/ER/tools/
[2] http://www.508checker.com/
[3] https://www.webaccessibility.com/


Figure 4.1: 508 Checker.

technology a means of empowering for people with disabilities. The validator is presented as an HTML page with a form in which to insert the URL of the Web page to be checked. At the end of the analysis, a downloadable report is provided after the user completes a form indicating his name, surname, company of which he is a part, telephone number and e-mail. The user can still see the results without downloading the report, just by clicking on the tabs that appear on the screen. The validator also provides a summary percentage on the total site compliance with the WCAG 2.0, 2.1 and Section 508 guidelines. An example can be seen in Figure 4.2. It supports WAI-ARIA and HTML formats.


Figure 4.2: Access Assistant Community.

### 4.1.3 Accessibility Insights for Web

Accessibility Insights for Web[4] is an open source validator of individual Web pages. It is an extension for Chrome or Microsoft Edge Insider, which provides two types of analysis: FastPass, a quick analysis highlighting only the most common problems, and Assessment, which checks more in detail if the site the user is consulting is in compliance with WCAG 2.0 level AA guidelines. The validator was released in March 2019 and supports the HTML format. In Figure 4.3 we can see an example of the resulting report.

[4] https://accessibilityinsights.io/docs/en/web/overview

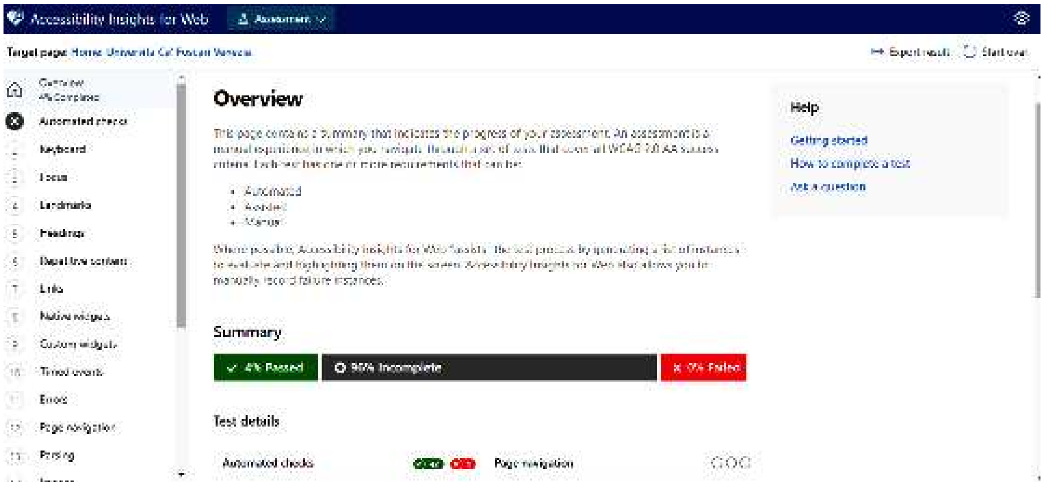


Figure 4.3: Accessibility Insight.

### 4.1.4 Cynthia Says

This validator[5] was released in April 2015. It checks the compliance of the single page present at the given url in the WCAG 2.0 guidelines and in Section 508. It is very simple to use, the user has to fill out the form by entering his email address, the URL of the site and the guidelines he wants verify. Supports CSS, HTML and image formats. The license is free. However, the user has to provide his email address and wait until the system sends the report. In Figure 4.4 we can see the principal page of this accessibility tool.

[5] http://www.cynthiasays.com/


Figure 4.4: Cynthia Says.


Figure 4.5: MAUVE Accessibility Validator.

### 4.1.5 MAUVE Accessibility Validator

This validator[6], released in July 2017 with a free license, supports CSS, HTML, XHTML. It checks if a single web page complies with WCAG 2.0 and 2.1 guidelines and the Stanca Law. The user sends the URL of the page in the appropriate field, choose the guidelines to check, eventually choose the device on which he wants to view the page and clicks the "Validate" button. It is also possible to choose to validate one's own file or code in the same way using the appropriate tabs. An example of the report producted can bee seen in Figure 4.5.

[6] https://mauve.isti.cnr.it/

### 4.1.6 Online-Utility.org Document Readability Test Tool

We decided to include this text validator[7] in the list, which can be very interesting for the purpose of understanding how complex a text can be: in fact, it does not control a Web page, but provides the complexity of a text inserted in the appropriate field. It checks the text through the Coleman Liau index, Flesch Kincaid Grade Level, ARI (Automated Readability Index), SMOG indices.
The report, as shown in Figure 4.6, is therefore a summary of the calculations performed on the text provided. In our case a test was made by inserting a part of Lorem Ipsum. It supports HTML format, the license is free and the release date is December 2014.

### 4.1.7 Tingtun Accessibility Checker

This open source validator[8] was released in January 2012. The interface is perhaps one of the simplest viewed, one can just enter in the appropriate field the address of the web page to be validated and press "Check". Supports HTML, XHTML and image formats. The report obtained shows which parts of the page do not comply with the WCAG 2.0 guidelines, first as a summary, then in detail. An example of the resulting report is shown in Figure 4.6.

[7] https://www.online-utility.org/english/readability_test_and_improve.jsp
[8] http://checkers.eiii.eu/en/pagecheck/

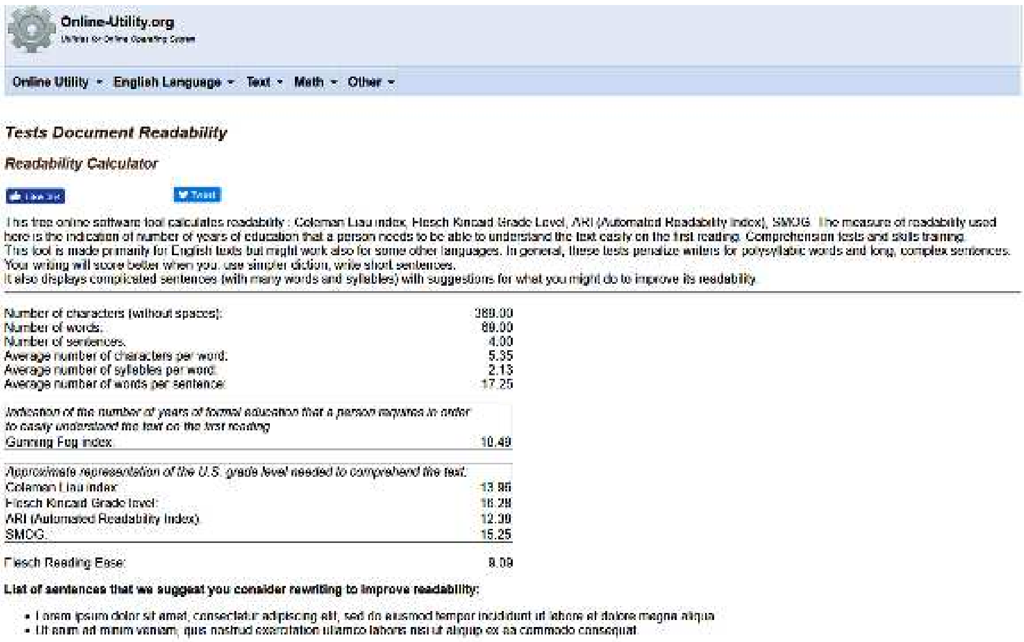


Figure 4.6: Document Readability Test Tool.

### 4.1.8 CSS Validation Service

This validator[9] has been made directly by W3C. It is open source and the year of release is 2009, although it is periodically updated. It allows the CSS validation of a Web page and for this reason it is very useful for developers. By entering the url in the appropriate field and choosing from the options the ones that most interest him, the programmer can receive a complete report on the CSS of the page. In Figure 4.8 we show both the form in the validator page and an example of the resulting report.

[9] http://www.css-validator.org/

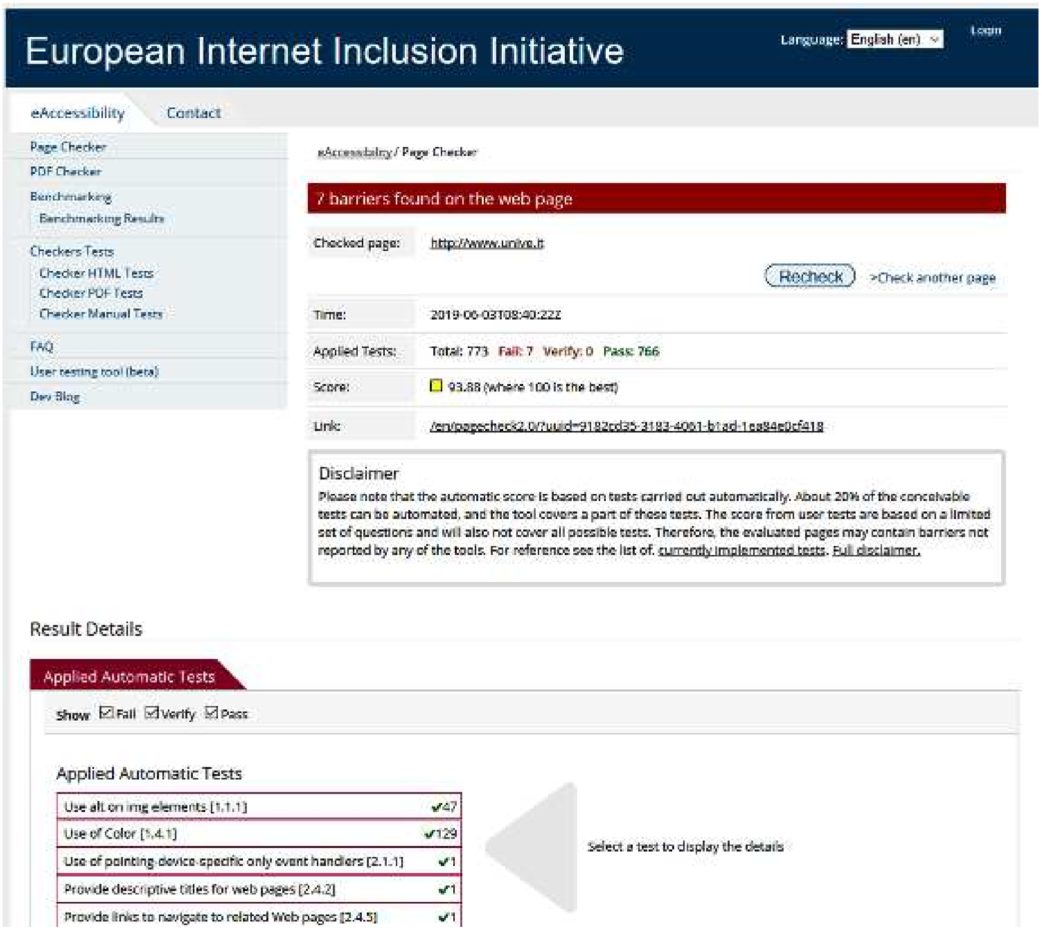


Figure 4.7: Tingtun Accessibility Checker.

### 4.1.9 AChecker

AChecker[10] is perhaps the most complete of the free and open source validators on the list. Released by Inclusive Design Research Center, a center

[10] https://achecker.ca/checker/index.php

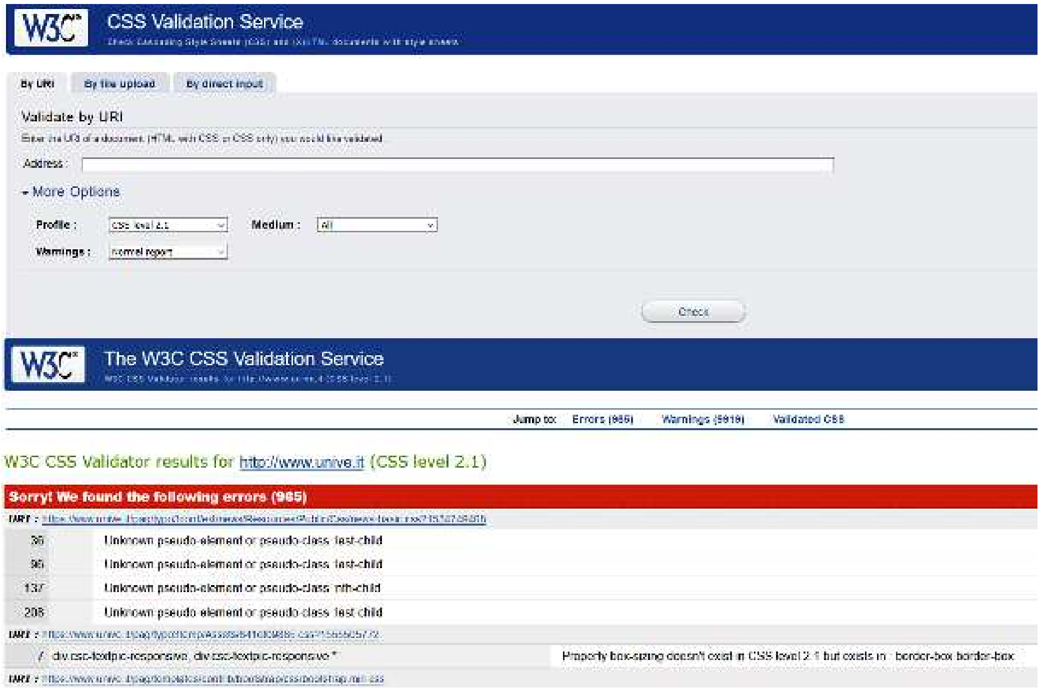


Figure 4.8: CSS Validation Service.

for the inclusion of the OCAD University of Toronto, even if it was born in 2008 it is still maintained. It checks individual Web pages, user-uploaded HTML files or parts of code. As shown in Figure 4.9, it is possible to choose between the WCAG 2.0 and WCAG 1.0 guidelines at various levels, Section 508, the BITV (which is the german equivalent to the Stanca Act) and the Stanca Act. A user can decide whether to validate only the HTML, only the CSS or both, and whether or not to show the code of the analyzed page. He can decide whether to see the report divided by guidelines or by line number. During the existing tools testing phase to decide what accessibility tool was the best starting point for our project, we realized that, due to one of the

latest updates, the page is processed also for the WCAG 2.1: this is one of the reasons that convinced us to use it as a base for our tool.
The report shows among the fundamental tabs *Known Problems*, which lists the problems that are certainly barriers to accessibility, *Likely Problems*, or problems that could be a barrier to accessibility but that require the revision of a human being to declare certainty , and *Potential Problems*, which are the problems that Achecker cannot identify without the intervention of a human being.

**Conclusion.** As we have seen, there are plenty of validators with different features. However not all of them are complete or updated: perhaps the paid tools for the companies are better, but there is no way of verifying it. What is certain is that none of these has been designed for people with Autism Spectrum Disorder and for this reason it is necessary to develop a dedicated validator.

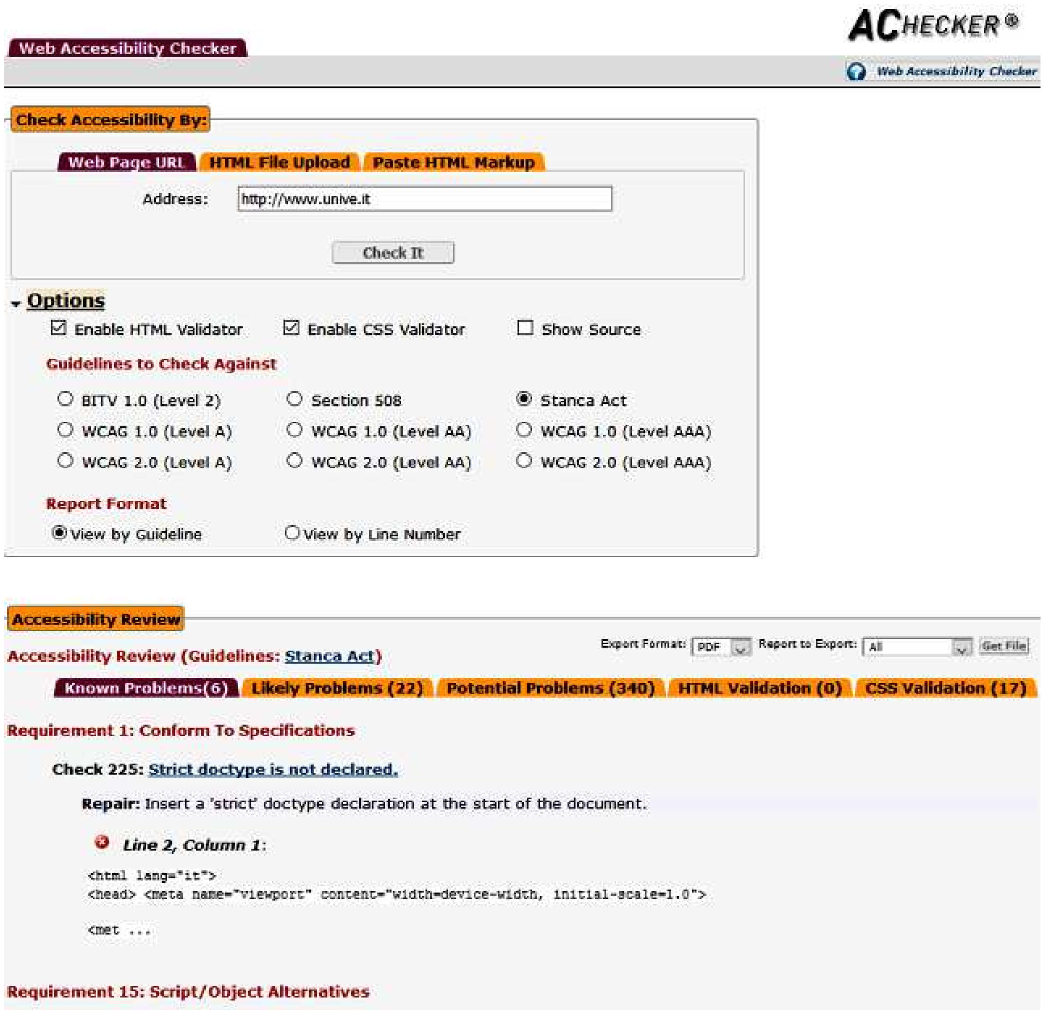


Figure 4.9: AChecker

# Chapter 5

# The Auditor tool

This chapter is entirely dedicated to the description of the Auditor tool. We describe its features and show some examples of code listings. We explain in detail the code, which is written in Php and Python languages, along with the stylistic choices we made.

## 5.1 The tool

It was decided to call the validator *Auditor*[1], from the crasis of Autism with Validator, so that its name recalls our goal.
The project is based on the previously presented Achecker tool, which is open source and, even if a bit slow, is the one that offers the most complete reports,

[1] https://tesimrossi.dais.unive.it/auditor/index.php

with all the guidelines indicated by WGAC 2.1 and the Stanca Act which we need. This slowness of the support tool has forced us to limit the validator to the analysis of individual pages, because for entire websites it would have had too much impact on the performances of the program. A complete listing of the code can be found here: https://github.com/Margherita842790/FinalThesis_Auditor.

Another Achecker problem is that, given that the css validation uses the W3C CSS Validation Service validator, which has been updated in the last few weeks and has a new layout, so Achecker cannot read the resulting report any more. It was decided to circumvent the problem by going directly to invoke the W3C validator, and reporting the problem to Achecker, trusting in their timely repairing modification.

### 5.1.1 Tool languages

Php and Python languages were used, due to their power in the field of Web content parsing. In particular, some scripts have been created in Python, which in turn are executed by a Php script. The validator is organized as follows. For the Php scripts, we have: `index.php`, `welcome.php`, `from.php`, `res.php`, `summary.php`, `help.php`, `problem.php`, `swrong.php`. For the Python scripts we have: `main.py`, `ckr.py`, `aut.py`, `aux.py`. From the main php page, the user can fill out the special form by entering the url of the page to be validated with or without including its protocol, then he can choose between

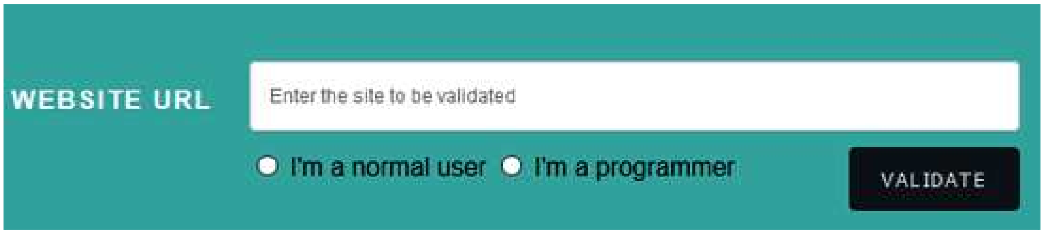


Figure 5.1: The form.

the expert user mode, which will lead to a detailed report, and the simple user mode, which will return a report that is easy to read even for those who are not in the sector.

We want to see now in detail the functionalities of the individual php pages, and then concentrate on the code written in Python, which can in all respects be considered the heart of the program.

### 5.1.2 index.php and welcome.php

`index.php` is the first page of the site. It contains the form in which the user can indicate the url of the Web page to be checked and if he is a normal user or a developer (see Figure 5.1). After the submit button is pressed it will be necessary to wait a few minutes.

From this page `welcome.php` is invoked: this page is not visible to the user, but it is necessary to be able to execute the python scripts.

The data are sent from the form in `index.php` using POST methodology, therefore the values are transmitted via HTTP between client and server in a “transparent” way for the user. In order to make everything work properly it is necessary to add the command ***ob_start()*** to the beginning of the script in `welcome.php`, so the buffering is activated and as long as it is active, it blocks the output and stores it in an internal buffer.

The data received by `welcome.php` are the provided url and the way in which the user wants to view the final report: with the ***isset()*** function it is checked whether the variable has been defined or not, for security reasons. It is also checked whether the url is malformed or non-existent: in that case the program returns a 0 and then an error page is invoked.

However, the command that interests us most is the exec command:

```
exec("./auditor/env/bin/python3 main.py '$scriptPath' '$url'");
```

Listing 5.1: exec command

this command allows us to run, via php, a program in another language, in our case Python. In order to make it working we need to write the path to the folder where our Python interpreter is located (in this case we have used a virtual environment created with Python 3.7.2).

**from.php** This page, like `welcome.php`, is not visible by the user, but it is useful because parts of code are copied by python in order to produce a pleasant graphic for the reports.


Figure 5.2: The navigation bar in res.

**res.php** This page presents the report with the graphics dedicated to the expert user. There is a navigation bar with 7 buttons, of which the first 5 recall the Achecker's cards (see Figure 5.2). The last two buttons are those dedicated to validation for autistic people. Clicking on the penultimate button, the entire validation appears according to the guidelines for autism. In particular, it is reported if *.gif* images are present or if there is any text into the images. Moreover, a text readability percentage is given and the most complex sentences are shown highlighted in different colors depending on the difficulty: red for very difficult sentences, orange for quite difficult sentences and yellow for difficult ones. At the bottom of the page it is also possible to see if colors were found to be unsuitable for people with Autism Spectrum Disorder, listed by item in which they appear.
The last button is dedicated to the summary sheet, the same that can also be found on the page dedicated to non-expert users. An example of the res.php page is shown in Figure 5.3.

**summary.php** This is the page dedicated to non-expert users. As previously mentioned, it presents a summary of the various inconsistencies found by the validator with respect to the guidelines approved for autism. It consists of a simple summary table in which we can find an overview of all the

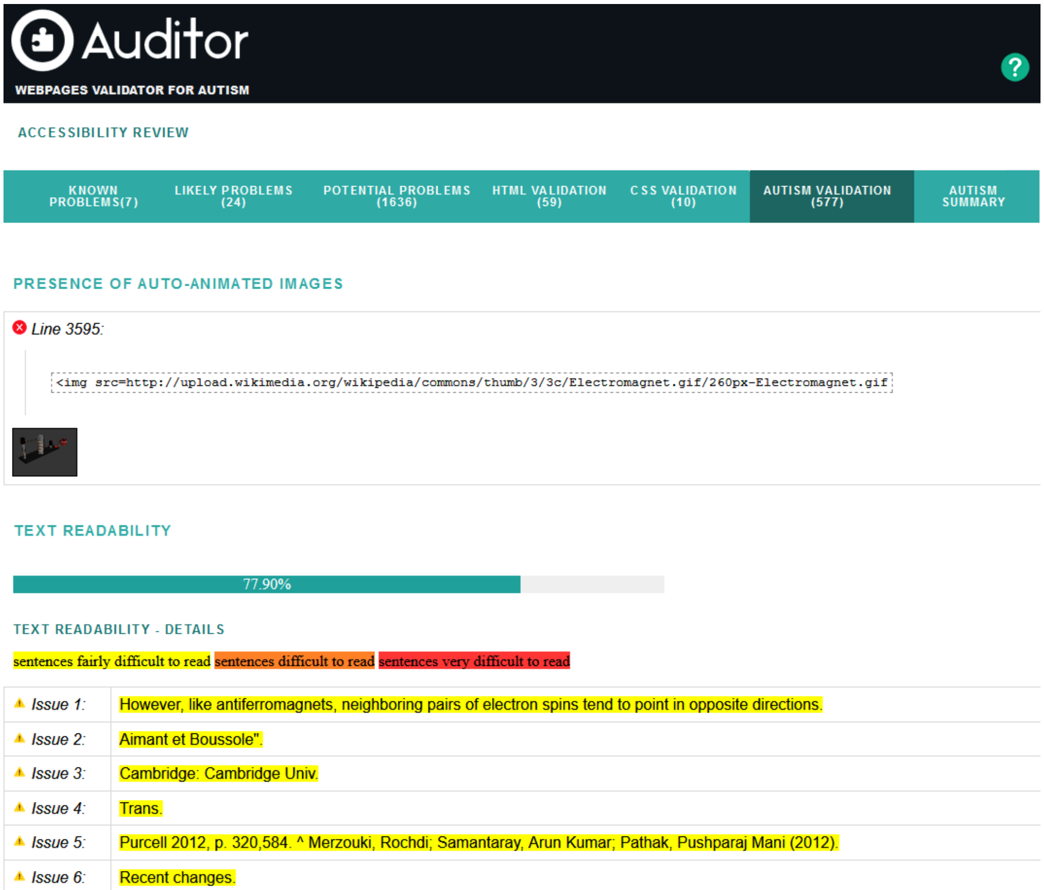


Figure 5.3: The res.php page.

possible difficulties that can be found, which presence is indicated in the right by a red cross, while the absence by a green tick. Further to the right there are some details. An example is shown in Figure 5.4.

**help.php, problem.php and swrong.php** `help.php` is the guide page where the user is explained how to use the validator. `problem.php` is instead the page that is invoked when there is an error, for example when for some

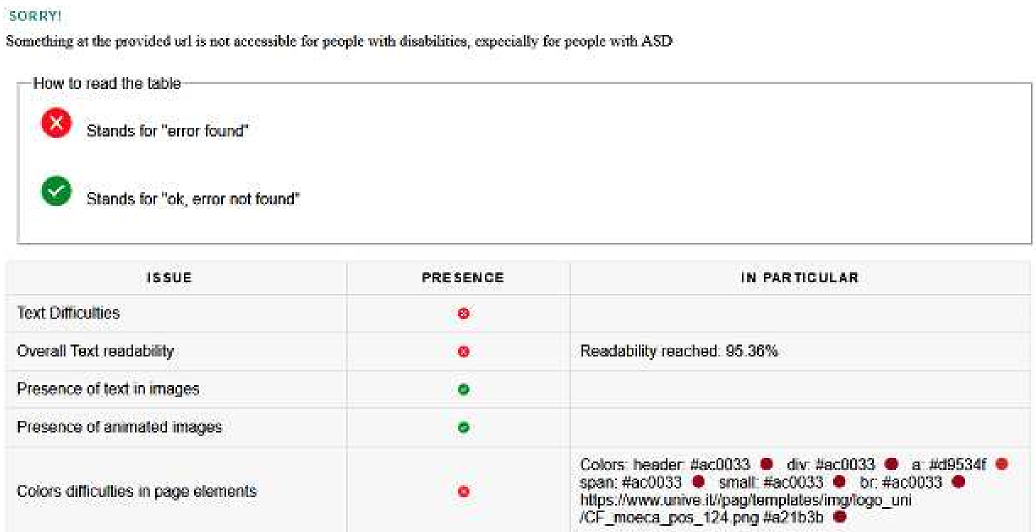


Figure 5.4: The summary.php page.

reason the program crashes. `swrong.php` is finally the page that is opened if a non-existent url is inserted.

### 5.1.3 The Python scripts and our choices for accessibility

Now we can talk about the actual program, the one that carries out all the calculations and addresses the different problems of accessibility and usability for people with autism spectrum syndrome.

First of all, we decided to focus only on some general guidelines: the validator, as well as parsing the report produced by Achecker, checks if there are moving images, if there are texts inside the images and if the colors are suitable for autistic people. We also analyse the text, providing a percentage of its readability and dividing it into sentences whose complexity is indicated by a specific color code.

Let us first see how the various problems were faced.

**Accessibility.** As the reader will have been able to see, we have decided to focus our implementation on some of the guidelines and studies we have discussed in the previous chapters.

The search for auto-animated images was important because the presence of such an element in a Web page can create a great inconvenience for people with autism.

We then focused on the problem of readability, because we are convinced that it is another key point of Web accessibility: it is certainly not easy to decide when a text is readable by anyone, which is why we have tried, through an in-depth study of the various readability indices, to choose the most common and suited ones.

Another aspect that we have decided to deal with is that of colors, which can really determine a problem in the consultation of a Web page by an autistic person: although establishing thresholds is not simple and has forced us to generalize a lot since not all the autistic people have the same disorders, it can be really useful for a developer or for relatives/teachers to know that the presence of a certain color could annoy the autistic person during navigation, since this could lead to decide to exclude a website from the consultation.

**Moving images.** Regarding the presence of moving images within a page, it was decided to parse the HTML or PHP code and go to look for all the images that presented the *.gif* or *.webp* format. To do this, we first look for all the images that are in the code. Later, since the extension is not always clearly visible, the python *imghdr* module is also used to determine the type with certainty.

```python
def get_content_type(url):
    req = urllib.request.Request(url, headers={'User-Agent': "Magic Browser"})
    data = urllib.request.urlopen(req).read()
    image_type = imghdr.what(None, data)
    if not image_type:
        try:
            image_type = Image.open(BytesIO(data))
        except:
            pass
    return image_type
```

Listing 5.2: img type evaluation

These images are then inserted into the final report, along with the line in which they are found. An example can be seen in Figure 5.5.

**Texts inside the images.** As for the text inside the images, keeping the list previously processed and having found the language used inside the site


Figure 5.5: Example of detected auto-animated image.

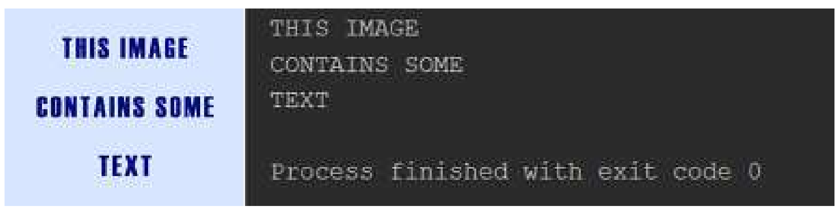


Figure 5.6: Example of image containing text and Tesseract result.

through the Python `TextBlob` library, the images are analysed one by one through `Opencv` module and then through `Tesseract` module.

`Opencv` is indeed a powerful library capable of processing images and videos and is often used for facial recognition, photo editing and more. Thanks to this library it is in fact possible to check if there is text in one or more images in the list.

`Tesseract`, in turn, is an optical character recognition tool that is useful for us to be able to read the found text and save the result for later, in order to calculate its complexity. In Figure 5.6 the result of processing Tesseract on an image is shown.

```
def text_in_img(url):
```

```
    headers = {'User-Agent': 'Mozilla 5.10'}
    request = urllib.request.Request(url, None, headers)
    response = urllib.request.urlopen(request)
    imagetext = ''
    try:
        if ax.is_grey_scale(io.BytesIO(response.read()))==False :
            arr = np.asarray(bytearray(response.read()), dtype=np.uint8)
            if arr.size != 0:
                img = cv2.imdecode(arr, -1)  # 'Load it as it is'
                if img is not None:
                    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
                    basedir = os.getcwd()
                    dirtess = os.path.join(basedir, 'tesseract.exe')
                    pytesseract.pytesseract.tesseract_cmd = dirtess
                    imagetext = pytesseract.image_to_string(gray, config='--tessdata-dir dirtess')
    except:
        pass
    return imagetext, url
```

Listing 5.3: Tesseract Use

**Text Analysis.** The text analysis was performed through the use of Python's ***Spacy*** library. This library, which is much more powerful than the ***TextBlob*** library mentioned above but which is very similar, allows us with its functions to calculate the complexity of a written text according to the different indices discussed in the previous chapters.

This open source library released by MIT implements statistical models of neural networks in English, Spanish, German, Portuguese, French, Italian, Dutch and Greek. In particular, with this library we can use operations such as Tokenisation, Lemmatisation, Part-of-Speech Tagging, Entity Recognition, Dependency Parsing, Sentence Recognition and many others. Its weakness can be found in the fact that it tends to treat all languages like English, and that is why, as we will see in the followings lines, we had to implement an additional function.

Since there's the possibility that a text is written inside an image, it was decided to calculate the complexity of the given webpage by making the average of the complexity of the texts present at the URL and the complexity of any texts previously found within the images, in order to have a complete report on the overall readability. Among the indices it was chosen to use, for the English language, the Flesh Reading Ease, as it is the most known and famous [33].

```
if lang == 'en':
            nlp = spacy.load("en_core_web_sm")
            doc = nlp(t)
```

```python
            sent_list = []
            for sent in doc.sents:
                sent_list.append(sent.text)
            read = Readability()
            nlp.add_pipe(read, last=True)
            for x in sent_list:
                docs = nlp(x)
                # grade.append(docs._.automated_readability_index)
                grade.append(docs._.flesch_kincaid_reading_ease)
```

Listing 5.4: Flesch Kincaid

On the contrary, given what was discussed in the chapter dedicated to the Natural Language Processing, for the Italian language the Gulpease index was considered better, but it was not implemented in ***Spacy***: it was therefore necessary to define a specific function for this index, following the equation previously presented.

The color code to be attributed to the individual sentences to give an idea of their complexity has been defined for the English language as follows:

- 0-40 sentences very difficult to read - red
- 41-60 sentences difficult to read - orange
- 61-80 sentences fairly difficult to read - yellow
- over 80 the sentence is considered simple (not colored nor visible)

TEXT READABILITY - DETAILS

sentences fairly difficult to read sentences difficult to read sentences very difficult to read

| | |
|---|---|
| *Issue 1:* | Scopri le nostre attività internazionali. |
| *Issue 2:* | I migliori servizi per accompagnare la carriera degli studenti e delle studentesse. |
| *Issue 3:* | Chiudendo questo banner o proseguendo nella navigazione si accetta luso dei cookie. |
| *Issue 4:* | in Italia Marie SkłodowskaCurie Fellowships. |
| *Issue 5:* | Dati necessari a comprendere a fondo il sistema climatico. |
| *Issue 6:* | Chiudi Informativa |

Figure 5.7: Example of text analysis.

while for the italian language:

- 0-50 sentences very difficult to read - red
- 51-70 sentences difficult to read - orange
- 71-80 sentences fairly difficult to read - yellow
- over 80 the sentence is considered simple (not colored nor visible)

The indices give us the legibility of a text and for this reason the scales range from the lowest to the highest score. An example is shown in Figure 5.7.

**Color's analysis.** Implementing the color's analysis was particularly difficult because we needed a way to formalize the Munsell system. Firstly, it was necessary to find all the colors within the webpage, which includes the css values of the different tags and also the colors within the images. As for the images, they were analysed through the use of `Opencv`, to transform them into a numpy array consisting of all the *coordinates* that define them. `Opencv` is a Python library which wraps the original C++ class and allows to perform image cleaning operations, to capture image features and more. In this way it was possible to find the colors of all the pixels of which the image is composed. Thus, since the image must be considered in its entirety, it was decided to determine the predominant color of the image by performing a K-means clustering.

To extract the color of the page tags, a parsing of the same was done once again and the function `value_of_css_property()` of `Selenium` was used to extract the css. Later on, all the tags that did not have in their css the "color" entry, nor the "font-color" or "background-color" or "background-image" were deleted. Only the relevant tags were processed.

At this point it is essential to talk about how the theme of color was treated and how the Munsell system was formalized.

Since this system is based on the combination of hue, value and chroma, which are nothing more than hue, saturation and lightness, it was thought that the most similar color model was the hsv model. This model in fact consists of hue saturation and value (which is the lightness), which correspond quite well to those of the Munsell system. The only care was to manually match the two color wheels, because compared to the Munsell system the hsv is slightly shifted, as we can see in Figure 5.8. After that, some thresholds have been established in order to decide if a color is suitable or not for an

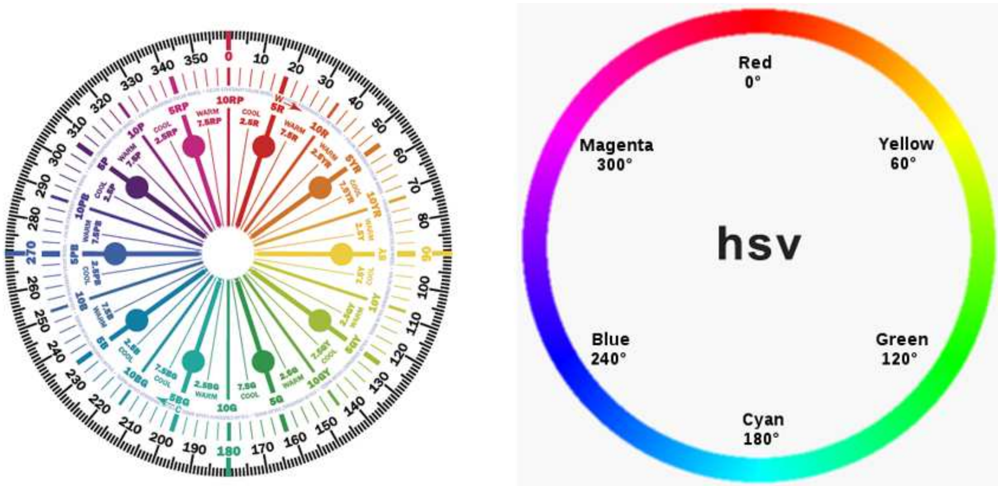


Figure 5.8: The Munsell wheel vs the hsv wheel.

autistic person. All the bright colors, the yellows, the reds and the pinks have been discarded. Instead, blues, not too bright greens and browns were kept.

```
def thresholds(h, s):
    s=s*100
    if s < 50:
        return True
    else:
        if(s < 70) and (10 <= h and h <= 45):
            return True
        else:
            if 80<=h and h<=285:
                return True
            else:
```

```
                    return False
```

Listing 5.5: color thresholds

### 5.1.4 Tool performance

How is it possible to do all these calculations without affecting the execution time too much? Unfortunately, basing on an existing validator which is already slow in itself, compromises must be done with the speed of execution, as we can see from Figure 5.9, which compares the total execution time for Achecker and our tool partial execution time, until it has obtained the Achecker report, before the autism guidelines analysis, while processing some example websites. Figure 5.10 compares Achecker total execution time in seconds and our tool's total execution time in seconds.

Another tool, which could not be dispensed with, and which has a minimal impact on the speed of the program, is *Selenium Webdriver*.
A Webdriver is a framework that allows us to automate some browser operations. In particular, Selenium is the Webdriver with which we send commands to the browser to perform the parsing operations. Selenium has been used in headless mode, i.e. without a graphical interface: this not only means that the user is not aware of its use, but it also makes everything faster.
We decided that it was more appropriate to choose Chrome Webdriver and use it in headless mode rather than choosing PhantomJS, which is a better-

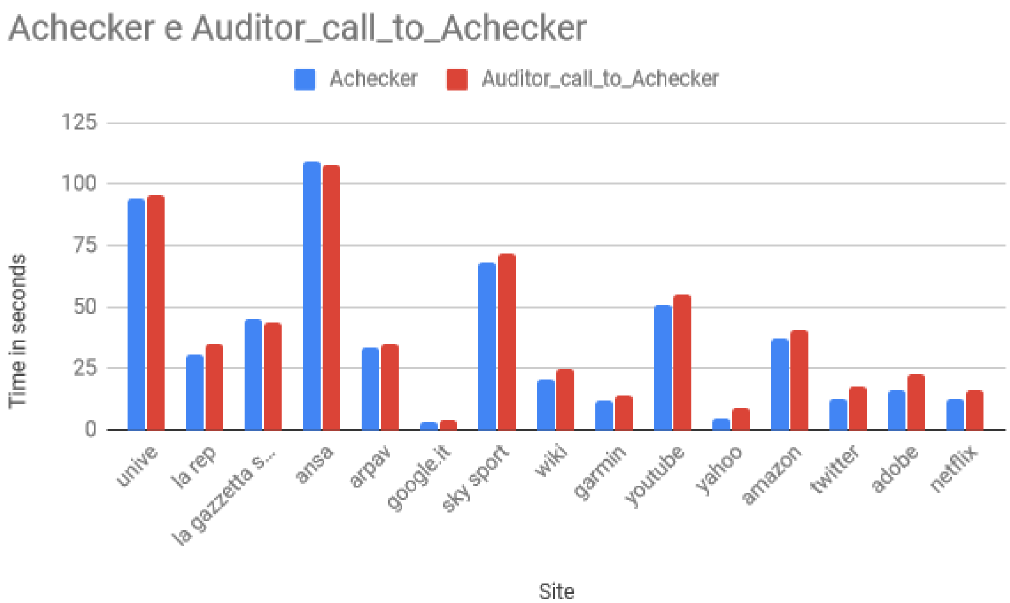


Figure 5.9: AChecker total execution time vs Auditor partial execution time while processing some example sites

known headless browser: the latter in fact recently proved to be less efficient, as can be seen online in numerous benchmarks; see for example Figure 5.11. We also tried to improve time performances by applying the technique of concurrency: it is about the code doing multiple things at the same time. As we know, dividing the workload into asynchronous processes can help a lot to avoid slowdowns.

Here all the functions that use Selenium Webdriver have been performed asynchronously.

```
p = Pool(4)
   p.apply_async(ck.sitecheck, (location, SITE_URL,))
```

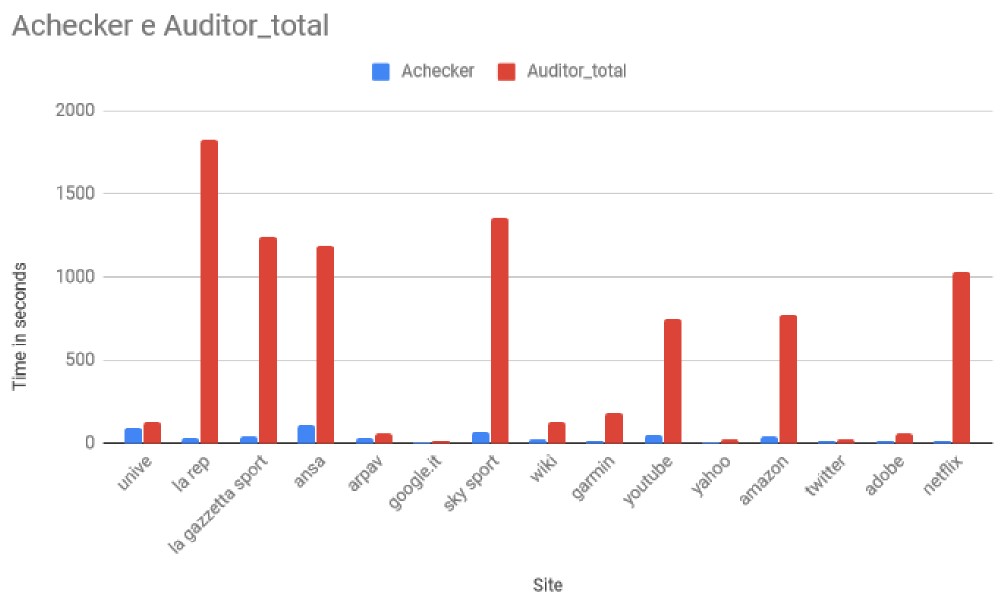


Figure 5.10: Auditor total execution time vs AChecker total execution time while processing some example sites

```
    p.apply_async(ck.sitecheck2, (location, SITE_URL,))
    source = p.apply_async(at.source, (SITE_URL,))
    tags_colors = p.apply_async(at.colors_tags, (SITE_URL, ))
    p.close()
    p.join()
```

Listing 5.6: Asynchronous Functions

The improvement using multiprocessing is not always significant due to the fact that when you use asynchronous execution, we do not create new threads, but we reuse those available in a thread pool. However, in this case

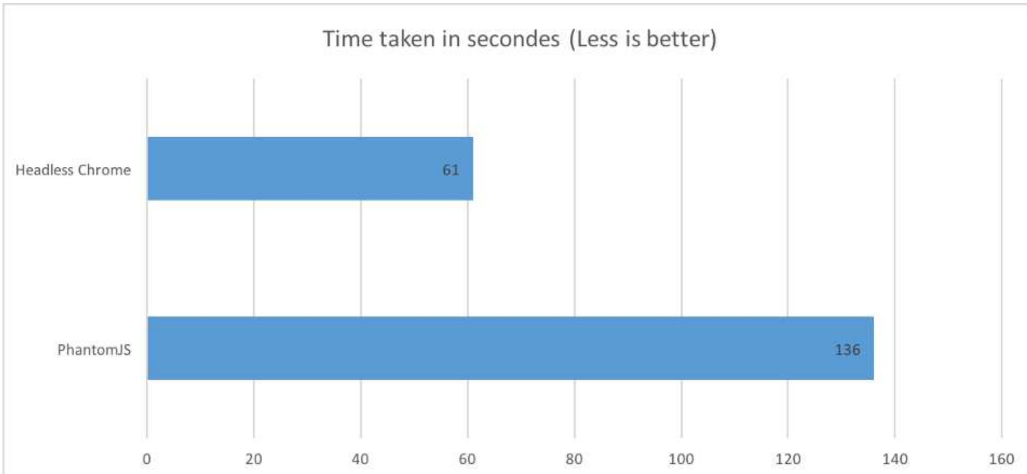


Figure 5.11: Headless Chrome Webdriver vs PhantomJS webdriver computed on the Rails framework's site in milliseconds [35].

we have had a little improvement in the performances of the tool. Moreover, as one can see in Figure 5.10, even if part of the slowdown is due to the call to Achecker, we cannot compare its performances with our tool's ones because the text analysis and the image research are very expensive, due to the fact that they require many http requests in order to be retrieved. To make a further improvement, one could go directly to the Achecker code, trying to parallelize it. Also, it might be useful to speed up Python scripts with Cython, a Python superset that translates code directly into optimized C / C ++.

**Conclusion.** Since, as we have seen in the previous chapter, many of the Web pages evaluation tools presented are not suitable to understand if a Web

content is friendly for autistic people, we decided to develop this tool, which is dedicated to developers and normal users to understand if a single Web page can be seen from people with ASD.
The time performances did not prove to be excellent, therefore it was necessary to make improvements through the use of headless browsers and multiprocesing, but it still needs some time to be executed. Even with these limitations, the validator obtained according to previously discussed precautions reflects our initial objectives, although the road to the improvement is always open.

# Conclusions

As we have seen, the topic of Web Accessibility for people with Autism Spectrum Disorder includes a wide range of knowledge which spaces from computer science to communication psychology and also to linguistics and acoustics.

Although many studies have sought new solutions in particular to address the disabilities given by Autism and in general on how to make the Web accessible to the people who suffer from it, there is still a widespread lack of application of the rules on the subject, which turns out to be very complex given the different skills to put in place.

Because of this complexity, it was difficult to develop this first attempt at automatic validation of Web pages for autistic people, and it is clear that the road is still long and still needs a lot of work mainly on understanding the disorder itself and then on how to deal with it through the tools provided by new technologies.

The project basically focuses on three Web Accessibility issues for people with ASD: the presence in a Web page of auto-animated images, the readability of a text within a Web content and the presence of colors that can be a source of discomfort for these people.

The most complicated argument turned out to be the calculation of the readability of a text, since it is calculated automatically by the use of mathematical formulas, and the fact that our work is done for people with different ages and symptoms, even if with the same diagnosis.

For what concerns colors, it was not easy to define the thresholds to report the unsuitable ones for these people: the fact that autistic people may have different symptoms gives the difficulty of having to choose what to report as annoying. It is not necessarily true that all autistic people prefers or are bothered by the same colors. Furthermore, it was not possible to find a way to decide the degree of contrast of an element with the background. It is not easy to divide the elements from their background, and we should be able to quantify how much contrast is needed to make an element visible.

However, we are convinced that the validator obtained is a good starting point for those who are interested in the subject, because it respects the objectives we had set ourselves.

Future work certainly involves the addition to our validator of a method to understand if the color of an object or of a text is sufficiently in contrast with the background. Another interesting improvement could be the development of a method to analyse the complexity of an image exactly as it was done here for the texts. Since we are still unable to decide whether an image is complex because it is composed of many elements or because it has many colors or for other reasons, it would be useful to formalize a method to specify what is meant by “complexity of an image”. To do this it would be advisable to create a dataset with the support of doctors and autistic people who can identify which images are easily understood and which are not, to then train an artificial intelligence and create dedicated algorithms.

It is also clear that a long testing phase will be necessary, in order to guarantee the effective validity of the tool. It would be useful to test the websites proposed in [28], to see if they would be still considered friendly for people with ASD. It could also be interesting to let developers and teachers/relatives to directly test the tool, allowing them to leave some feedbacks. Finally, these results could be combined to understand what can be changed and improved.

# Bibliography


[1] S. de Boer-Ott, D. E. Griswold, B. S. Myles, S. E. Byrd, J. B. Ganz, K. T. Cook, K. Otten, J. Ben-Arieh, S. A. Kline, L. G. Adams, et al. *Autism spectrum disorders: Interventions and treatments for children and youth*. Corwin press, 2004.

[2] O. Bogdashina. *Sensory perceptual issues in autism and asperger syndrome: different sensory experiences-different perceptual worlds*. Jessica Kingsley Publishers, 2016.

[3] N. Chomsky. *Syntactic structures*. Walter de Gruyter, 2002.

[4] M. Coleman and T. L. Liau. “A computer readability formula designed for machine scoring.” In: *Journal of Applied Psychology* 60.2 (1975), p. 283.

[5] E. Dale and J. S. Chall. “A formula for predicting readability: Instructions”. In: *Educational research bulletin* (1948), pp. 37–54.

[6] A. Dattolo and F. Luccio. “Accessible and UsableWebsites and Mobile Applications for People with Autism Spectrum Disorders: a Comparative Study.” In: *ICST Trans. Ambient Systems* 4.13 (2017), e5.

[7] R. M. Foxx. “Applied behavior analysis treatment of autism: The state of the art”. In: *Child and adolescent psychiatric clinics of North America* 17.4 (2008), pp. 821–834.

[8] O. Gaggi, P. Manzoni, C. Palazzi, A. Bujari, and J. Marquez-Barja. *Smart Objects and Technologies for Social Good: Second International Conference, (GOODTECHS 2016), Venice, Italy, November 30 – December 1, 2016, Proceedings, LNICST*. Vol. 195. Jan. 2017.

[9] C. Gold, T. Wigram, and C. Elefant. “Music therapy for autistic spectrum disorder”. In: *Cochrane Database of Systematic Reviews* 2 (2006).

[10] M. Grandgeorge and N. Masataka. “Atypical color preference in children with autism spectrum disorder”. In: *Frontiers in psychology* 7 (2016), p. 1976.

[11] L. Kenny, C. Hattersley, B. Molins, C. Buckley, C. Povey, and E. Pellicano. “Which terms should be used to describe autism? Perspectives from the UK autism community”. In: *Autism* 20.(4) (2016), pp. 442–462.

[12] G. Kielhofner. *Conceptual foundations of occupational therapy practice.* FA Davis, 2009.

[13] P. Legrenzi. *Fondamenti di Psicologia Generale*. ilMulino, 2016.

[14] P. Lucisano and M. E. Piemontese. “GULPEASE: una formula per la predizione della difficoltà dei testi in lingua italiana”. In: *Scuola e città* 3.31 (1988), pp. 110–124.

[15] C. D Manning and H. Schütze. *Foundations of statistical natural language processing*. MIT press, 1999.

[16] J. Nielsen. *Usability Engineering*. 1994.

[17] V. K. Pampuro. “Incorporating Augmentative and Alternative Communication Usage In Functional Therapy For Autism Spectrum Disorder: A Case Study”. In: (2014).

[18] N. Pavlov. “User interface for people with autism spectrum disorders”. In: *Journal of Software Engineering and Applications* 7.02 (2014), p. 128.

[19] P. Righini. *L’acustica per il musicista, fondamenti fisici della musica*. G.Zanibon, 1978.

[20] S. J. Russell, P. Norvig, and E. Davis. *Artificial Intelligence : A Modern Approach*. 3rd ed. Pearson, 2010, pp. 860–918.

[21] O. Sacks. *The Man Who Mistook His Wife for a Hat*. Adelphi, 1985.

[22] D. R. Simmons, A. E. Robertson, L. S. McKay, E. Toal, P. McAleer, and F. E. Pollick. “Vision in autism spectrum disorders”. In: *Vision research* 49.22 (2009), pp. 2705–2739.

[23] M. Sipser. *Introduction to the theory of computation*. Cengage learning, 2012.

[24] A. Tomatis. *Pourquoi Mozart?* Fixot, 1991.

[25] S. Wang, M. Jiang, X. M. Duchesne, E. A. Laugeson, D. P. Kennedy, R. Adolphs, and Q. Zhao. "Atypical visual saliency in autism spectrum disorder quantified through model-based eye tracking". In: *Neuron* 88.3 (2015), pp. 604–616.

[26] J. Whipple. "Music in intervention for children and adolescents with autism: A meta-analysis". In: *Journal of music therapy* 41.2 (2004), pp. 90–106.

[27] T. Winograd. "Understanding Natural Language". In: *Cognitive Psychology* 3.1 (July 1972), pp. 1–191.

[28] B. Yu, M. Murrietta, A. Horacek, and J. Drew. "A Survey of Autism Spectrum Disorder Friendly Websites". In: *SMU Data Science Review* 1.2 (2018), p. 8.

# Sitography


[29] Amazon. *The top 500 sites on the web*. 2019. URL: https://www.alexa.com/topsites.

[30] ASDEU. URL: http://asdeu.eu/.

[31] Autismspeaks. URL: https://www.autismspeaks.org/.

[32] CDC. *Key Findings from the ADDM Network*. URL: https://www.cdc.gov/ncbddd/autism/addm-community-report/key-findings.html.

[33] Suzane Davies. *The Flesch Reading Ease and Flesch-Kincaid Grade Level*. 2017. URL: https://readable.com/blog/the-flesch-reading-ease-and-flesch-kincaid-grade-level/.

[34] Italian Government. *Law 9 January, n. 4*. 2004. URL: https://www.gazzettaufficiale.it/eli/id/2004/01/17/004G0015/sg.

[35] *Headless Chrome vs PhantomJS Benchmark*. 217. URL: https://hackernoon.com/benchmark-headless-chrome-vs-phantomjs-e7f44c6956c.

[36] *Munsell Color System*. URL: https://munsell.com/.

[37] NAS. URL: https://www.autism.org.uk.

[38] *PECS Autism Communication Picture Cards for Children with ASD*. 2017. URL: https://www.autismag.org/news/pecs-autism-communication-picture-cards-for-children-with-asd/.

[39] M. Ritter. “Inclusive Design isn’t rocket science”. In: (2018). URL: https://blog.prototypr.io/inclusive-design-isnt-rocket-science-7ec3423a9b55.

[40] TEACCH. *TEACCH program*. URL: https://teacch.com/.

[41] *The TOMATIS Method*. URL: https://www.tomatis.com/en.

[42] National Health Services UK. *Signs of autism*. URL: https://www.nhs.uk/conditions/autism/signs/#.

[43] *Usability*. Standard. International Organization for Standardization, 2018. URL: https://www.iso.org/obp/ui.

[44] W3C. URL: https://www.w3.org/WAI/.