Pict2Text 2.0. Translating messages with pictograms into text

Final project
Course 2020–2021

Autors
Gasan Mohamad Nazer
and
Veronika Borislavova Yankova

Directors
Virginia Francisco Gilmartín
and
Susana Bautista Blasco

Bachelor's Degree in Software Engineering
Faculty of Informatics
Complutense University of Madrid

Pict2Text 2.0. Translating messages with pictograms into text

Bachelor's Degree Final Project in Software Engineering

Autors
Gasan Mohamad Nazer
and
Veronika Borislavova Yankova

Directors
Virginia Francisco Gilmartín
and
Susana Bautista Blasco

Convocation: *June 2021*

Bachelor's Degree in Software Engineering
Faculty of Informatics
Complutense University of Madrid

November 12, 2020

Contents

Chapter 1

Introduction

This chapter will display the motivation behind this project, its objectives, and the software development methodology used during its elaboration. In the end, we present the document structure with the different sections within it and a brief description of them.

1.1. Motivation

Communication is a pillar in interpersonal relationships, a fundamental need in our society. Unlike most people, for some, this action requires a lot of effort. Their differences create an uncrossable barrier and almost impossible human connection using the traditional way of communication. To remove this barrier an alternative approach should be used - pictograms. These graphic images, representing an object or a concept, have helped a lot to establish an initial communication channel between people with special needs and the rest. Although there are specialists trained to work and teach this unique language, for the majority of people, these pictograms are unknown and their purpose is not fulfilled.

To completely include pictograms and transform the communication between people with disabilities and those without, it is needed to harvest the capabilities of modern technologies, and create software. Software, which can "translate" pictograms into natural language - Spanish.
Currently, multiple tools allow transforming natural language into pictograms. The initial version of "Pict2Text" goes a step further and does the reverse action - translating the pictograms into natural language. But to realize this, it requires a sentence to be written using pictograms that are selected manually searching for every one of them. This is not good enough as the people with disabilities who need the pictograms, cannot write and search the specific word so that it can be matched to the pictogram they want. This

problem can be solved, giving those people the option to upload a picture of a sentence constructed with pictograms. With the next generation of "Pict2Text", version 2, we aim to reduce the exclusion of those people with special needs from society and break the communication barrier, providing the aforementioned functionality.

The beneficiaries of this software will be the above mentioned two groups, people with communication problems and those without them. In the first group, we include individuals who use pictograms to communicate, those with cerebral palsy, autism, or any other kind of cognitive disability. The second group is for all the others, users of the natural language. This project is created to help both of the previously mentioned groups, integrating the first one as a normal part of the society, able to communicate freely, expressing their needs, feelings, desires, and the second not only as assistants for the needs of the first group but also as friends and equal. In this modern, 21st century world, we all should be equal independently of if we can or cannot use the natural language in our communication.

1.2. Goals

The main goal of this project is the creation of an image classification machine learning model capable of distinguishing single or multiple pictograms from a given image.

Integrate a pictogram classification model as new functionality and extend the previous version of "Pict2Text".

Use services-oriented application architecture, constructing web services, and/or microservices.

Use industry standards during the whole process.

Consolidate and expand the knowledge acquired during the period of acquiring the Software Engineering Degree.

1.3. Software development methodology

Given our initial inexperience with the problem we wanted to solve, we decided to use an agile methodology. Unlike the traditional methodologies, that are more suitable when you have clearly defined requirements, and follow a linear approach - initiation, planning, execution, monitoring, and closure, agile methodologies are more flexible and focus less on initial planning. The small iterations will allow us to receive feedback before the end of the final project, which makes it easier to correct mistakes on time.

After further consideration, we selected Kanban as the most suitable methodology for us. The product is delivered continuously, and changes are allowed during the whole process, and no estimation of the tasks is needed. That will increase our flexibility and productivity and will decrease the effect of our lack of experience in estimating.

During every meeting, the tutors will give us the tasks that we have to finish until the next one, including their priority. The date for the next meeting also will be decided during the current one, as the sprint may vary between two and four weeks, depending on the given tasks. We decided to have the following columns:

- To do. Here we will put the tasks given to us by the tutors in the order they determine.
- In progress. This column contains the tasks we are currently working on. All tasks in that column will have a particular person assigned. As we are using Kanban, we have limited the WIP (Work In Progress) of the column to 4 assignments to avoid doing more tasks than we can reasonably manage. If more tasks come from the column "Testing" or "Ready for review", increasing the number of tasks more than the given WIP, some of them will be moved to the column “On hold”.
- On hold. Here we will put the activities that we can’t continue at the moment. The reason behind this could be that they are waiting for another task to finish. When the task could be continued, it is, once again, moved to “In progress”.
- Testing. In the case of programming assignments, the testing will include code review and automated and/or manual testing. If documentation is in this column, the person who didn’t write the particular part will read and correct it. When the testing is finished, the task is moved to "In progress" if a bug is spotted or the task is not finished, or to “Ready for review” in the opposite case.
- Ready for review. This column contains the tasks we have finished but are not yet reviewed by our tutors. The tasks in that column will be reviewed during the next meeting and the tutors will decide which of them will be moved to the column “Done” and, in case of a task that is not completed, to the column “In progress”.
- Done. The last column contains the tasks that are finished, reviewed, and approved by the tutors.

1.4. Document structure

The document is structured as follows.

Chapter 1(Introduction) where we have covered the motivation, the goals, and the methodology for the project.

Chapter 2(State of the Art) describes pictograms and existing tools for their translation, covers the state of the first version of "Pict2Text", the functionalities we will include to increase its usefulness, and the general idea of the machine learning algorithms and models we will be using.

Chapter 2

State of the Art

In this chapter, we have briefly defined Augmentative and Alternative Systems of Communication (section 2.1) and pictograms (section 2.2). More information regarding the first two sections you can find in the documentation of Pict2Text version 1. In continuation, we will explain Pict2Text version 1, its functionality, and its issue in section 2.3 and 2.4 respectively. In section 2.5. we will present the machine learning models we have analyzed to solve our problem and their characteristics.

2.1. Augmentative and Alternative Systems of Communication (AAC)

The Augmentative and Alternative Systems of Communication (AAC) is a way of communication that does not include verbal speech or writing. It is used to assist people with disabilities such as autism, cerebral palsy, dual sensory impairments, genetic syndromes, intellectual disability, multiple disabilities, hearing impairment, disease, stroke, head injury, and etcetera. AAC helps those people to express their needs, thoughts, and desires and improves their abilities to interact with others. AAC is often being personalized to match the needs of its user. It includes different systems of symbols: graphic (pictures, drawings, pictograms, words or letters) and gestural (mimicry, gestures, or hand signals).

2.2. Pictograms

Pictograms are ideograms that contain a meaning represented through their resemblance to a physical object, activity, etcetera. They can be used to form a sentence (image 2.1), which makes them suitable for people who cannot use written or verbal speech. As pictograms are not universal, various

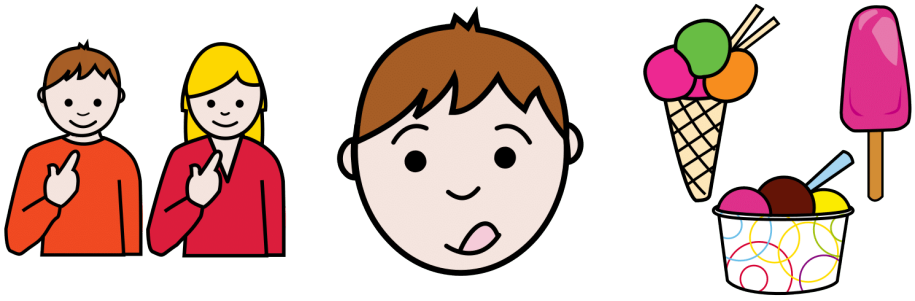

Figure 2.1: Pictograms making the sentence "We like icecream"

systems exist, such as Blissymbolics, CSUP, ARASAAC, and more.

2.2.1. Blissymbolics

Blissymbolics[1] is an ideographic language consisting of several hundred basic symbols, each representing a concept, which can be combined to generate new symbols that represent new concepts (image 2.2). Blissymbolics characters do not correspond to the sounds of any spoken language and have their use in the education of people with communication difficulties.

2.2.2. CSUP

The Communication System Using Pictograms (CSP) uses pictograms to support interactive non-verbal communication. This system uses pictograms not only for objects but with events as well. It is designed in a way that it can be used between a person with a disability and a non-disabled person, child and adult, people speaking different languages, and so on.

2.2.3. ARASAAC

The ARASAAC[2] project was created in 2007, and it currently consists of more than 30.000 pictograms, including complex pictograms with already constructed phrases, in more than 20 languages, differentiating between singular and plural, as well as gender differentiation (image 2.3). Verbs come with a different pictogram for every conjugation, and the tense is determined by pictograms representing yesterday, today, and tomorrow. The pictograms are separated into five groups - coloured pictograms, black and white, photographs, and sign language videos and pictures. With a wide variety of pictograms, ARASAAC is free to use an internationally recognized

[1] https://en.wikipedia.org/wiki/Blissymbols
[2] https://arasaac.org/

pictogram system, used by many people in numerous countries. Taking in consideration the beforementioned and the fact that Pict2Text vesion 1 works with the pictograms they obtain from their website, we decided to use them as well.

2.3. Pict2Text version 1

As described previously, Pict2Text version 1 is the initial state and the base of our project. The first version of this project is a web application that permits the translation of pictograms to natural language- Spanish.

2.3.1. User perspective

Using the user interface we can write and search a specific word from the ARASAAC pictogram database and display it into a panel on the right part of the web page. After that, we can include the chosen one into the pictogram sentence panel, from where later the message with pictograms will be translated into natural language. The following images and descriptions present a simple flow of actions a user can do to achieve the above-mentioned behavior. When entering the website[3] the user can see on the left part, a big panel, the pictogram sentence panel, with a caption "Pictograms" above it, and a button "Traducir" below it. On the right part, an input box with a caption "Nombre del picto", and a button "Buscar" on the left of it. (See picture 2.4)

To translate pictograms to natural language, the user should first search for a pictogram. To do that, they should write the world they are looking for in the input box of the right side and click the button "Buscar". In image 2.5 it is shown a search of the word "Hombre" and the corresponding pictogram.

Having the pictogram, the next action needed is to include it into the left panel with pictograms by clicking the button "Añadir". In image 2.6 it is shown the pictogram corresponding to the word "Hombre" included in the pictogram sentence panel.

Repeating the previous steps with other words, a sentence can be formed. In image 2.7 it is shown a translation of a sentence written with the pictogram corresponding to the searched words "Hombre", "Comer", "pizza".

2.3.2. Engineering perspective

The core of Pict2Text is the API of ARASAAC. It provides the searching mechanism used to match words to pictograms, the graphical images of pictograms, and additional information about them.

[3] https://holstein.fdi.ucm.es/tfg-pict2text

A generator of grammatically correct phrases in Spanish was needed. In version 1 of this project was used SimpleNLG, a Java library for natural language generation. This library permits the creation of simple and complex phrases. To do that, it requires sentence structure- subject, verb, adjectives, gender, and number (singular or plural) of every word in the formed sentence. With this information, SimpleNLG can generate grammatically correct sentences.

Spacy is the tool that gives the previous word characteristics. It is a python library with a high accuracy used for advanced natural language processing.

As all of the different functionalities from the project were implemented as web service, most of them in Python, the team of Pict2Text version 1 have decided to use the framework Django for integration and intercommunication between them.

For the front-end of the project, it was used Angular. As the website itself is a SPA(Single-Page Applications), which needs to respond fast, a framework like Angular fulfills this performance requirement.

2.3.3. Pict2Text version 1 issue

Although the first version of Pict2Text translates the pictograms into natural language, it requires the user to manually select the pictograms they want to use in the construction of their sentence. Writing the words is impossible for people with disabilities who need pictograms to communicate if it was not, they would have used the natural language in the first place.

This problem can be solved, giving those people the option to upload a picture of a sentence constructed with pictograms. The functionality we are building will be able to separate the different pictograms from the original image and later translate the phrase using the implementation in Pict2Text version 1.

2.4. One-shot learning algorithm

One-shot[4] is a machine learning algorithm that aims at object categorization based on one or a few training examples. As shown in image 2.8, the algorithm is based on comparing the similarity between pictures. Given two images, if the objects are the same, the neural network returns a value that is smaller than a specific threshold. In the other case, the returned value will be higher than the threshold.

[4]https://bdtechtalks.com/2020/08/12/what-is-one-shot-learning/

Figure 2.2: Blissymbolics pictogram system

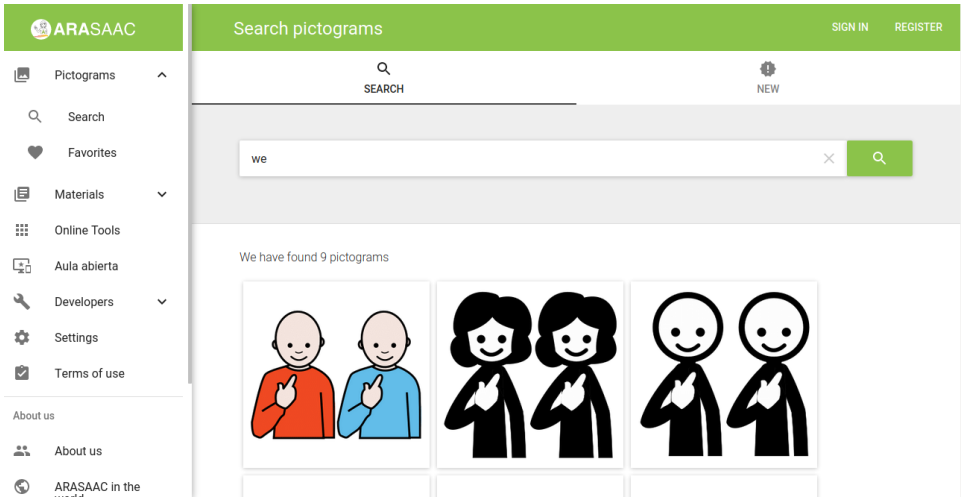

Figure 2.3: Pictograms in ARASAAC website

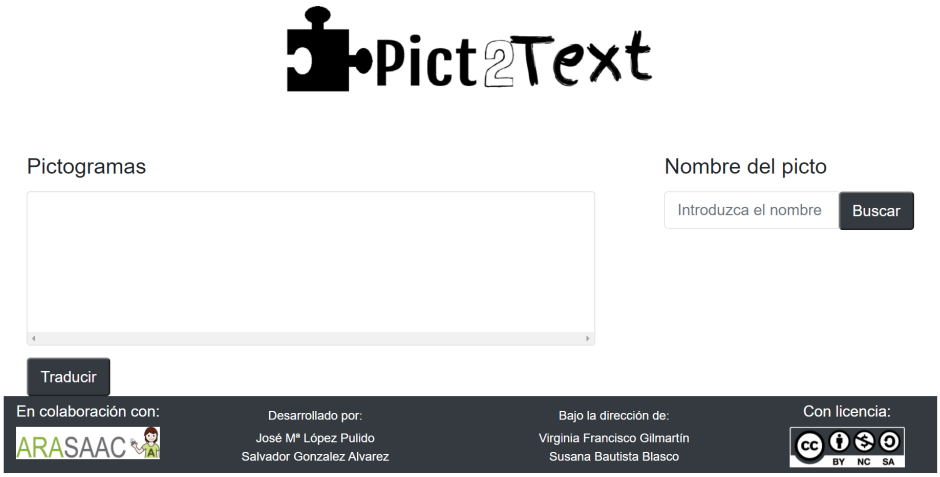

Figure 2.4: Pict2Text version 1 website

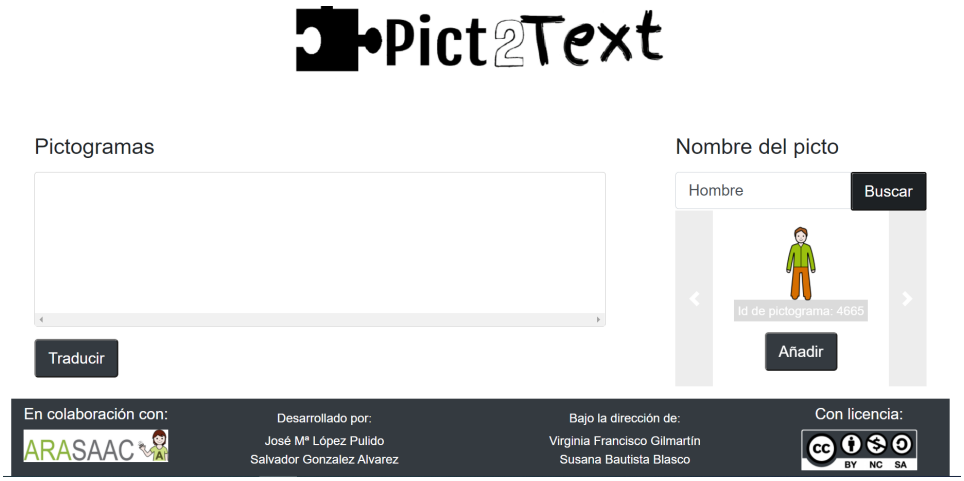

Figure 2.5: Searching the word "Hombre"

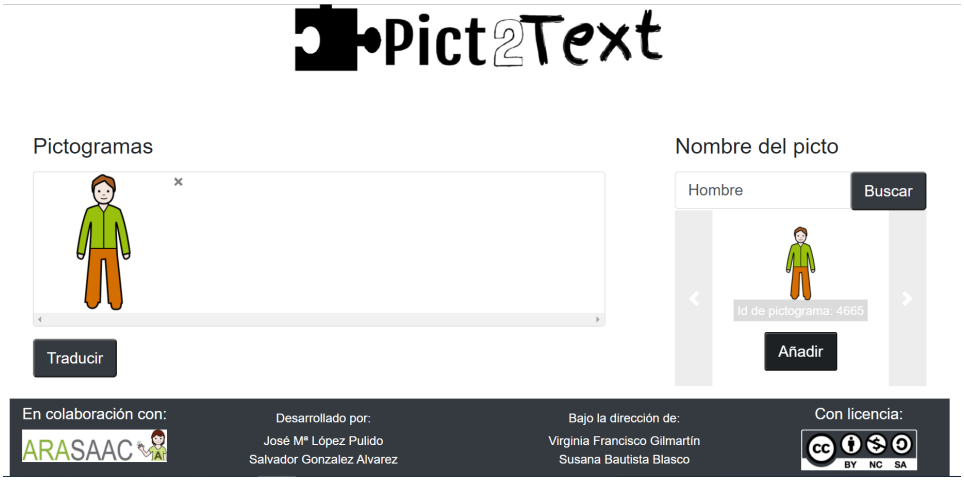

Figure 2.6: Adding the pictogram "Hombre" to the pictogram sentence panel

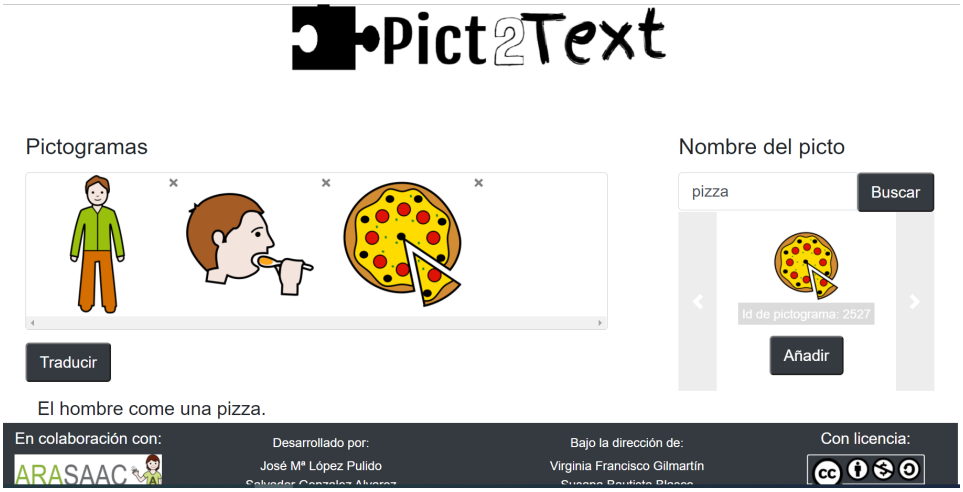

Figure 2.7: Translating the sentence "El hombre come una pizza."

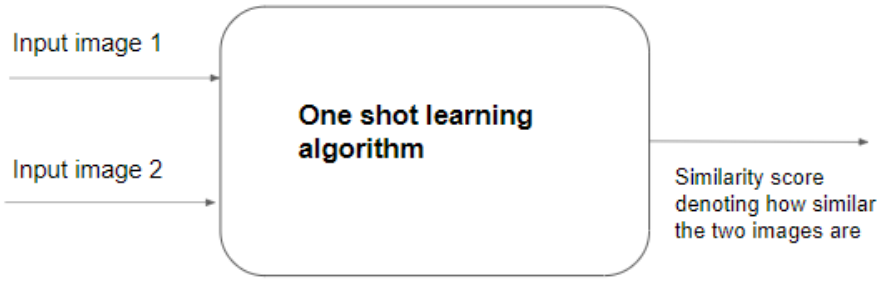

Figure 2.8: One-shot learning algorithm model