Master of Science in Informatics at Grenoble (MOSIG)
Specialization : Data Science and Artificial Intelligence (DSAI)

Automatic and Optimized Communication Grid Generation from Artificial Intelligence Techniques

Hugo Bantignies

28th June, 2022

Research project performed at Laboratoire d'Informatique de Grenoble (LIG)
Team : *Groupe d'Étude en Traduction Automatique/Traitement Automatisé des Langues et de la Parole (GETALP)*

Under the supervision of:
Didier Schwab and Benjamin Lecouteux

Defended before a jury composed of:
Massih-Reza Amini
Georges Quénot
Jean-Pierre Chevallet

June 2022

Abstract

In Augmentative and Alternative Communication (AAC), pictogram communication grids are manually conceived and frequently improved by speech-language pathologists. However, to generate such grids in an automatic way is challenging : the various organizations of pictograms and the use of a grid depend mainly on the user's requirements or abilities. Previous promising works represent an interesting starting point that we want to pursue [5][28]. We establish a pipeline that generates random pictogram grids from raw texts corpora and relied on a genetic algorithm to optimize the efficiency of their usages. To adapt the algorithm to pictogram grids, we introduce new genetic operators regrouping a crossover and various mutations. We also define an hybrid cost to evaluate the efficiency of a grid that the process will try to minimize, based on the pictogram distances within the grid and their semantic similarities. We present a Natural Language Processing (NLP) pipeline to preprocess non-annotated corpora to obtain sentences that are as close as possible to the language of pictograms. Additionally, we develop PictoGriz, a visualization tool to efficiently get an overview of large pictograms grids and to be able to analyze them. The extensive evaluation we build shows some of our optimized grids can be approximately as effective as popular grids such as PODD. Furthermore, it reveals how the vocabulary of the dataset used within the optimization pipeline impacts the ability of building elemental or complex pictogram sequences.

Résumé

En Communication Améliorée et Alternative (CAA), les grilles de communication de pictogrammes sont construites à la main et fréquemment améliorées par des orthophonistes. Malheureusement, générer ces grilles de manière automatique est une tâche complexe : les méthodes pour organiser les pictogrammes sont variées et leurs utilisations dépendent principalement des besoins de l'utilisateur ainsi que de ses facultés. Quelques travaux ont commencé à traiter de ce sujet et ont présenté des résultats encourageants, formant une base solide que nous souhaitons approfondir [5][28]. Nous proposons une pipeline permettant de générer des grilles de pictogrammes aléatoires à partir de corpus de textes et de les optimiser grâce à un algorithme génétique. Pour adapter correctement notre algorithme aux grilles de pictogrammes, nous mettons en place de nouveaux opérateurs génétiques tels que différentes mutations et un nouveau croisement. Nous définissons un coût hybride pour évaluer l'éfficacitée d'une grille que le traitement va essayer de minimiser. Ce coût est calculé en fonction des distances entre les pictogrammes au sein de la grille et de leurs similarité sémantique. Nous présentons une pipeline de Traitement Automatique des Langues (TAL) afin de prétraiter des corpus non annotés et obtenir des phrases se rapprochant d'un language à base de pictogrammes. Nous avons aussi développé PictoGriz, un outil de visualisation pour analyser et avoir une vision globale de plus larges grilles de pictogrammes. Nous avons mené une évaluation approximative pour montrer que nos grilles optimisées peuvent parfois atteindre la même efficacité que les grilles populaires comme le PODD. De surcroît, cette évaluation révèle l'impact du vocabulaire présent dans les bases de données sur la capacité à former des séquences de pictogrammes.

Acknowledgement

I would like to express my sincere gratitude to Mr. Didier Schwab for offering me the opportunity to work on this project and for believing in myself and my capacities to deal with this subject. The assistance provided by Mr. Benjamin Lecouteux was beneficial both for the internship and the report. I am also grateful to the GETALP team, especially to Cécile Macaire for helping me on datasets and documentations and to Mr. Gilles Sérasset for DBNary setup. I would also like to thank Mr. Renaud Blanch for his Information Visualization lectures and his pieces of advice that helped me develop PictoGriz.

Special thanks to Pauline Maurel for reading through this report and helping me correct some of the English mistakes. I wish to extend my special thanks to my supportive laboratory friends and particularly to Théo Nommay, Arthur Van-Schendel, Ivan Deryabin and Calvin Massonnet.

It was a pleasure to work for LIG. I improved my research skills and gained experience and I am grateful for having contributed in both AAC and NLP domains.

Hugo Bantignies.

Contents

Abstract **i**

Résumé **i**

Acknowledgement **ii**

1 Introduction **1**
1.1 Context and Motivations 1
1.2 Problem Statement 2
1.3 Proposed Automatic Optimized Pictogram Grid Generation Solution 2
1.3.1 Approaches 2
1.3.2 Results and Analyze 3
1.4 Contributions 3
1.5 Report outline 4

2 State-of-the-Art **5**
2.1 Augmentative and Alternative Communication 5
2.1.1 Unaided AAC systems 5
2.1.1.1 Gestures 5
2.1.1.2 Vocalizations and Natural Speech 6
2.1.2 Aided AAC systems 6
2.1.2.1 Objects 6
2.1.2.2 Photographs and drawings 6
2.2 Pictogram Grid Communication Systems 6
2.2.1 Pictograms 6
2.2.1.1 Advanced pictograms 7
2.2.2 Pictogram Grids and Organization 7
2.2.2.1 Activity grid displays 8
2.2.2.2 Taxonomic grid displays 8
2.2.2.3 Semantic-Syntactic grid displays 8
2.2.2.4 Pragmatic Organisation Dynamic Display 8
2.2.2.5 Other displays 9
2.2.3 Grid Cost Evaluation 9
2.3 Genetic Algorithms 9

2.3.1 Optimized Pictogram Grid Generation using GAs 10
2.3.2 Adaptive Genetic Algorithms 10
2.4 Natural Language Processing contributions for AAC 10
2.4.1 Automatic pictogram generation from spontaneous speech 11
2.4.2 Predictive Models for AAC 11
2.4.2.1 Statistical Approaches and Models 11
2.4.2.2 PictoBERT for Pictogram Prediction 11

3 Automatic Generation of an Optimized Pictogram Grid 13
3.1 Overview of the Optimization Pipeline 13
3.2 Pictogram Grid Structure 13
3.2.1 Grid Hierarchy 14
3.2.1.1 Pictogram 14
3.2.1.2 Page 14
3.2.1.3 Grid 14
3.2.2 Tree Structure and Navigation within the Grid 14
3.2.2.1 Page and Tree dimensions 14
3.3 Evaluation of a Pictogram Grid 15
3.3.1 Distance Cost 16
3.3.2 Coherence Cost 16
3.4 Genetic Algorithm for Pictogram Grid Optimization 17
3.4.1 Population and Parameters 17
3.4.2 Genetic Operations 17
3.4.2.1 Selection 17
3.4.2.2 Crossover 17
3.4.2.3 Mutations 18
3.4.3 Multiprocessing and Parallelization 19
3.5 PictoGriz : a Pictogram Grid Visualization Tool 20
3.5.1 Overview of the tool 20
3.5.2 Implementation and Details 21
3.5.3 Interactions 21
3.5.4 Future Improvements 21

4 Dataset 23
4.1 Dataset and Preprocessing 23
4.1.1 TCOF Dataset 23
4.1.2 Preprocessing and Vocabulary 24
4.1.2.1 Pipeline for Corpus Preprocessing 24
4.1.2.2 Language Model for Lemmatization 25
4.1.2.3 Undesired Words Filtering 25
4.1.2.4 Adult-Child TCOF Results 26

5 Experiments and Evaluation 27
5.1 Automatic Optimized Pictogram Grid Generation 27
5.1.1 Configurations and Parameters 27
5.1.2 Coherence Coefficient Variation and Displays 28

5.1.2.1 Text Corpora . . . 28
5.1.2.2 Distance Grids . . . 28
5.1.2.3 Coherent Grids . . . 30
5.1.2.4 Hybrid Grids . . . 30
5.2 Evaluation with Existing Standards . . . 34
5.2.1 Evaluation Description . . . 34
5.2.2 Results . . . 34
5.2.2.1 Approximate Evaluation with a French PODD 15 . . . 34
5.2.2.2 Diverse Dataset . . . 36
5.3 Discussion of Results . . . 37
5.3.1 Small-Scale and Large-Scale Optimization . . . 37
5.3.2 Pictogram Grid Evaluation with Standards . . . 37

6 Conclusion and Perspectives **39**
6.1 Conclusion . . . 39
6.2 Perspectives . . . 40

A Appendix **41**
A.1 One page structure and cutting . . . 41
A.1.1 Similarity-based cut . . . 41
A.1.2 Proximity Score cut . . . 41
A.2 Detailed Adaptive Mutations Process . . . 42
A.3 AnimalTexts Sample . . . 42
A.4 Execution Time for the Grids Optimization . . . 43
A.5 Page Examples from the Optimized Grids . . . 43
A.6 AT_{Hybrid} Grid in Details . . . 44

Bibliography **47**

— 1 —

Introduction

1.1 Context and Motivations

"*A picture is worth a thousand words*" (Frederick R. Barnard), speech is not the only way to communicate and some people are unable to use it. Communication is a human need and it should be accessible by all.

Augmentative and Alternative Communication (AAC) regroups all the ways a person can use to communicate without talking, such as body languages, drawing or by using a support. Those modes of communication can be aided or unaided. AAC has a prominent place in the Natural Language Processing (NLP) domain and is mainly helpful for people with production or understanding of speech impairments.

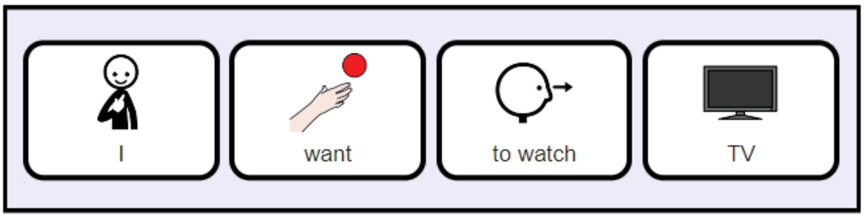


Figure 1.1: Example of a pictogram sequence built on the AugCom platform [32].

An interesting aided AAC technique is the use of pictogram grids also called Pictogram Grid Communication Systems (PGCS). In such systems, words are encoded into pictograms (iconic pictures) and the user will communicate by selecting pictograms to form a sequence. Figure 1.1 illustrates an example of a pictogram sequence. The organization of such grids and how pictograms are arranged mainly depend on the user's needs and abilities. Nowadays, pictogram grids are hand-drawn and improved daily by specialists such as audiologists, which may be a long and painful process. However, there exists no process to automatically update or generate a pictogram grid that would be optimized for any user. Preliminary works started to explore this field but it remains essential to be improved and analyzed in more details [5][28].

1.2 Problem Statement

The efficiency of the pictogram grid is mostly based on how fast a user can build its pictogram sequences with ease. Every pictogram grid has different purposes, ways of usages and should correspond to the targeted person. We consider that an optimized pictogram grid has to be as efficient as possible while respecting a particular organization (displays, pages, etc.) and constraints.

Starting from a raw text corpus, our objective is to generate optimized pictogram grids automatically by using machine learning algorithms and taking into account the diversity of options and AAC challenges. In the end, we want to solve a minimization problem to optimize the pictogram grid. Then, we need to explore different approaches to automatically evaluate the efficiency and ability of a pictogram grid. We also have to find approaches to build datasets that are suitable for the language of pictograms. Finally, we have to consider how to analyze large pictogram grids in a short period of time because it remains an important aspect for our work.

1.3 Proposed Automatic Optimized Pictogram Grid Generation Solution

This section describes the different approaches we propose and summarize the results we obtain and the task evaluation we conduct.

1.3.1 Approaches

As expressed previously, our goal is to generate optimized pictogram grids automatically. We want the generated grids to be efficient and subject to constraints and to organizations of the pictograms.

First, we explore an automatic approach to evaluate the efficiency of a pictogram grid called the cost of the grid. From existing display types, emerged two organization categories, mostly influenced by the syntax and the semantics. We present an hybrid formula to compute the cost $Cost(G)$ of a pictogram based on the distance between pictograms $Dist(G)$ within the grid and their coherence $Cohe(G)$ (semantic similarities between pictograms within different pages). A coherence coefficient α is injected to the cost computation, allowing variation between syntactic and semantic impact on the organization and efficiency.

Then, we design an adapted genetic algorithm to minimize the pictogram grid cost $Cost(G)$ we presented that corresponds with the fitness. The algorithm handles a population of initially randomized pictogram grids that evolves during a succession of generation. We conceive different genetic operators to be suitable with the pictogram grids. The crossover is mostly based on the information on the position of the pictograms and it transmits an amount of pictogram position to the next individual with a given rate (CIR). The mutations aim to swap, duplicate or delete pictograms in the grid without degrading its core vocabulary.

To collect the vocabulary of the grid and evaluate its cost for minimization, a text corpus is essential. We come up with a preprocessing pipeline to simplify texts or transcriptions from natural language to fit with the language of pictograms. First, we use a language model to lemmatize the sentences and then, we filter the undesirable words such as determinants or transcriptions errors that are not representative of pictogram sequences.

Finally, to analyze in a short period of time large pictogram grids, we develop a visualization tool (PictoGriz), displaying grid global views and allowing us to navigate or inspect the grids in an efficient way due to its proposed interactions.

1.3.2 Results and Analyze

Our first experiment aims to test the optimized pictogram grid generation by varying the coherence coefficient with different sized corpora. We observe that small-scale optimizations are faster and return efficient grids that are close enough to the targeted displays. However, for larger grids, results are favourable but the organizations seem to slightly differ from our expectations. We discuss on the many reasons that could be at the origin of those unexpected phenomena.

Then, we conduct an evaluation to situate and compare our generations with existing popular pictogram grids. The obtained results are an approximation, since, complexity choices, we apply a transformation of the structure on the standard grids. Nevertheless, this evaluation shows the efficiency of small grids and how close the larger pictograms grids are with those used by specialists. Finally, an analysis of the coverage of the grids vocabulary with several texts corpora - including more or less complex sentences - show that the quality of the language in the dataset directly impacts the ability to form pictogram sequence by using the optimized grid.

1.4 Contributions

The major contributions from this work are described as follows :

- A new hybrid evaluation cost for a pictogram grid implying distance and semantic similarities between pictograms.

- The design of a genetic algorithm that is able to optimize pictogram grids by minimizing their cost. It also regroups the genetic operators we defined such as crossover and mutations.

- An NLP preprocessing pipeline to adapt texts from a raw corpus to the language of pictograms and user's needs.

- PictoGriz, a visualization tool to get an overview of large pictogram grids and efficiently analyze them.

- A task evaluation to compare different pictogram grids based on their cost and their vocabulary.

1.5 Report outline

Chapter 1 establishes the context and motivations and introduces our work as the results. Chapter 2 presents an exhaustive state of the art to situate pictogram grids in the AAC and NLP domains. Chapter 3 regroups the different approaches and the pipeline we introduced to automatically generate optimized pictogram grids. It also includes the presentation of PictoGriz, a grid visualization tool we developed. Chapter 4 explains the dataset choices we have made and the detailed pipeline to preprocess text corpus. Chapter 5 covers the different tests and experiments we performed and discusses them. It also presents the evaluation we conducted to compare pictogram grids. Chapter 6 concludes about our several contributions and suggests perspectives and alternatives for future works.

— 2 —
State-of-the-Art

2.1 Augmentative and Alternative Communication

The American Speech-Language-Hearing (ASHA) - a professional and scientific American association regrouping audiologists, speech-language pathologists, scientists, etc. - provides a definition for AAC : *"Augmentative and alternative communication (AAC) is an area of clinical practice that supplements or compensates for impairments in speech-language production and/or comprehension, including spoken and written modes of communication."* [35]. AAC systems are regrouping multiple different techniques that can be aided or unaided. A system is unaided when the user does not need external support to communicate such as gestures or signs. An aided AAC may imply a variety of supports, involving high-technology or not. Phones, computers, books are a few examples of options for aided systems. It is necessary to take in consideration the ability of the user to interact with the support to communicate. AAC is mostly individual based, assessments will be required to find an adapted technique and the best options for each individual. More details about AAC assessment can be found in [2]. Therefore, AAC is not exclusive to people with impairments, everyone can benefit from it. Following sections show the existing unaided and aided AAC systems used by many people.

2.1.1 Unaided AAC systems

The unaided AAC systems do not require physical support to help the user. There exists different techniques, generally based on body language or vocalizations.

2.1.1.1 Gestures

A popular unaided method is the use of gestures to communicate. Those are movements of a part of the body such as the fingers, the eyes, the arm, etc. to manifest a meaning or an intent. For instance, by using facial expressions, a person can express an opinion or answer a close-ended question. The main advantage of gestures, which is related to the unaided mode, is that there is no need of any equipment and it can be used everywhere. However, this method is limited by the amount of possible messages the person can produce and the details of the information. Furthermore, the use of gestures depends a lot on the individual's motor skills.

The gesture techniques regroup also sign systems like the American Sign Language (ASL). However, as ASL is a thorough and complex language, it is not always appropriate for AAC. Nevertheless, there exists simplified sign systems like *Baby Signs* or *Manual Signs* [2].

2.1.1.2 Vocalizations and Natural Speech

Individuals who cannot produce a complete speech because of their impairments, may emit vocalizations to communicate. Similarly to onomatopoeia, the producing sounds will have no meaning out of context, but will be understood depending on the situation. The main limitation in an unaided mode is that the partner of the individual has to know and remember what is the meaning of each vocalization to be able to understand it.

2.1.2 Aided AAC systems

An AAC system is considered as aided when the user needs an external support helping communication. People may employ several aided methods, using low-tech or high-tech objects.

2.1.2.1 Objects

To communicate using physical objects belongs to the aided AAC systems. The individual will select one of several objects to describe a situation or answer a question. An example of situation can be the following : the individual can answer the question *"How are you going to travel for your vacation ?"* by pointing toys like a plane or a car. Nevertheless, this method is limited by the set of available objects and how relevant they are to what the user wants to express.

2.1.2.2 Photographs and drawings

As objects may be bulky and inconvenient in many situations, the use of photographs or drawings to communicate is an efficient aided alternative. Individuals will show photographs or drawing to express what they want. Before, people were using real pictures, from books or portfolios, but the process could be slow and rather limited. Nowadays, the support may be electronic such as tablets or phones and include applications that benefits from the web or large databases to provide an important collection of pictures. The use of pictograms is a popular drawing aided AAC and will be discussed in more details in the next section.

2.2 Pictogram Grid Communication Systems

A popular aided AAC system is the Pictogram Grid Communication System (PGCS). The main purpose of a PGCS is to allow a person to communicate by showing and building sequence of pictograms. We describe what pictograms are and how PGCS are designed.

2.2.1 Pictograms

Pictograms are iconic pictures encoding words or meaning to facilitate the understanding. Figure 2.1 shows an example of a few pictograms representing different Olympics sports and we can identify them effortlessly : *Rugby, Sailing, Shooting, Skateboarding*.

Such as Egyptian's hieroglyphs, pictograms form a complete formal language and are used to communicate by building pictogram sequences. However, this system's main limitation is the richness of the vocabulary. As a word is represented by a graphic symbol, it may be

Figure 2.1: Samples of pictograms from the Tokyo 2020 Summer Olympics [34].

challenging to find correct pictograms for complex words. How to schematize words such as *idealism, philosophy, etc.* to be understood by everyone ?

2.2.1.1 Advanced pictograms

Before, we saw that for each word, we associate one pictogram to represent it and it may be rather arduous. There are more complicated and advanced pictograms allowing more freedom for the user. Polysemic pictograms may be used to represent synonyms with a single icon. For instance, the pictogram *[Me, Myself, I]* will contain one image for all those meanings. Pictogram sentences encode a part of a sentence, an expression or a question with a unique graphic symbol. An idea of a pictogram sentence for the English expression *To beat around the bush* is to draw a bush and draw a person beating the ground around it. A new kind of pictogram also emerged, called Meta-Pictograms in literature. The concept is to act on communication and influence it by using special pictograms[18].

2.2.2 Pictogram Grids and Organization

A pictogram grid is a set of different pages containing pictograms. Each page is composed of x rows and y columns of pictograms. Figure 2.2 is a 5x3 page example of a pictogram grid.

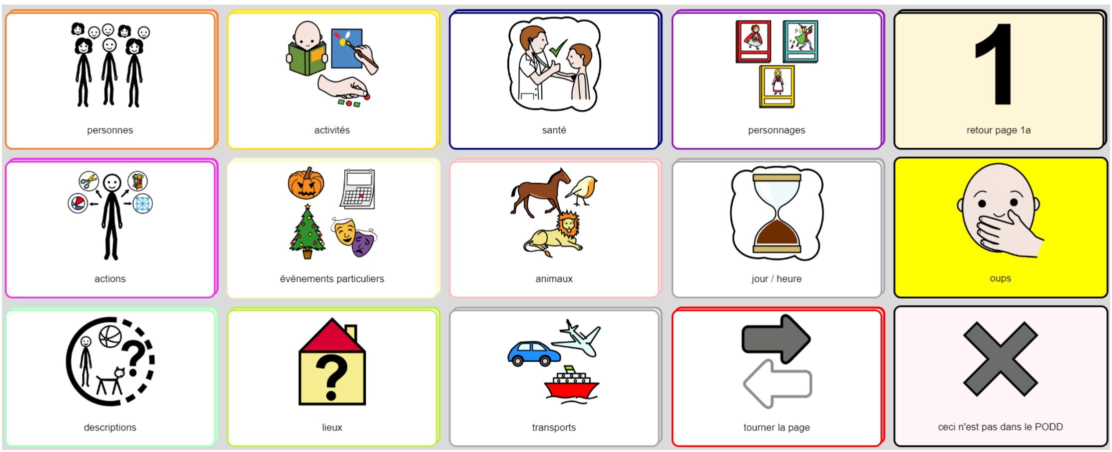


Figure 2.2: Example of a pictogram page from a french PODD 15 on the AugCom platform. [32]

We can navigate from one page to another following the links given by the directory pictograms. We can make a short analogy with a file system, where the files are the pictograms and the pages are the directories. On the Figure 2.2, *"personnes"* is a directory pictogram and by selecting it, we will be redirected to the page *"personnes"*. The displacement of the pictograms within a page can differ, depending on the purpose of the grid and it is really seminal for the efficiency of the grid and it is less time consuming. For instance, one ideal solution is to regroup pictograms having similar meaning (cheese, mouse, cat, ...) in a same page. The user will quickly select "mouse" and "cat" to build a pictogram sequence like *"Cat eat mouse"*. In the following sections, we present a typology regrouping different techniques to organize pictograms. Those displays are mostly influenced by two linguistics elements : syntax and semantics.

2.2.2.1 Activity grid displays

The schematic or activity display is a way to regroup pictograms related to the same event or activity. This organization is relatively efficient in a conversation about one particular event. Those displays are also called contextual displays [18]. The use of context is of great important to describe a situation and to facilitate communication. This display makes it possible to have the same pictogram in different pages for efficiency purposes. However, this may increase significantly the number of pictograms and the size of the entire grid.

2.2.2.2 Taxonomic grid displays

Inspired from taxonomy in the field of biology field, the science of classification, taxonomic display will regroup pictograms belonging to the same category. Generally, hypernyms will be categories and will gather more specific pictograms. *Vehicles* is a category and some associated pictograms can be *Car, Plane, Train*. As contextual displays, we may have a pictogram included in multiple categories. However, the number of occurrences of each pictogram will be smaller than for a contextual display.

2.2.2.3 Semantic-Syntactic grid displays

Previous displays, activity or taxonomic, are generally based on the proximity of the thematic between different words. Semantic-Syntactic display is a way to gather pictograms by their usage to build sentences. In a same grid, we may not have related words by their meanings, but because one word is commonly used with another in a natural language like English or French. In sentences starting with *"I love"*, the verb "love" can be used before words with opposite meanings such as *"cats"* or *"to ski"*, so those words may appear in the same display.

2.2.2.4 Pragmatic Organisation Dynamic Display

The Pragmatic Organisation Dynamic Display (PODD) system has been introduced by Gayle Porter in 2007 [21]. The PODD is a popular display and is often used by the speech-language pathologists across the world. This system is pragmatic ; it is mostly based on communication functions. The use of communication will always fulfill a need or a purpose. There exists four main different functions : Information, Motivation, Control, Emotional expression. For instance, the information function represents communication to share information or knowledge, whereas the control function is for exercising authority. The PODD facilitates the navigation between pages with different methods such as page numbers on the pictograms or operational commands like "*Go back to page 1"*. The size of the PODDs is not fixed. A PODD 9 will be constituted of pages containing 9 pictograms while pages of a PODD 20 will contain 20 pictograms.

2.2.2.5 Other displays

We just described the major pictogram grid displays in previous sections. However there is an infinite number of possibilities depending on various parameters and depending on the purpose of the grid.

The alphabetical display will regroup pictograms by their first letter. This can be efficient for a small grid.

The chronological display is organized so that pictograms have a temporal order. An action A occurring at time t will be placed before an action B coming at time $t+1$ in the grid.

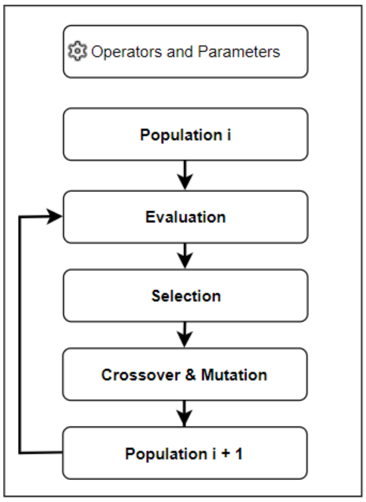
2.2.3 Grid Cost Evaluation

Knowing if a grid is efficient and optimized is very challenging because of the variety of purposes and individuals. Nowadays, there exists very few automatic ways to evaluate a pictogram grid. Previous works present a formula to evaluate a grid given a sentence or a text considering two different costs : the movement and the selection [5]. The movement and the selection cost mainly depend on the way to interact with the pictogram grid. For instance, an eye-tracker involves a short time to navigate within the grid, but a long time to select a pictogram. A weighted graph is extracted from the pictogram grid, representing the euclidean distance between each pair of pictograms belonging to the same page. Then, a Dijkstra algorithm is applied on this graph to find the shortest pictogram path we need to construct the input sentence. The main drawback of this method is the complexity to evaluate a grid, and this makes it unusable for evaluating large-sized grids and applying a genetic algorithm. Except for this first approach, automatic evaluation of pictogram grids is quite recent and there are barely any existing solutions.

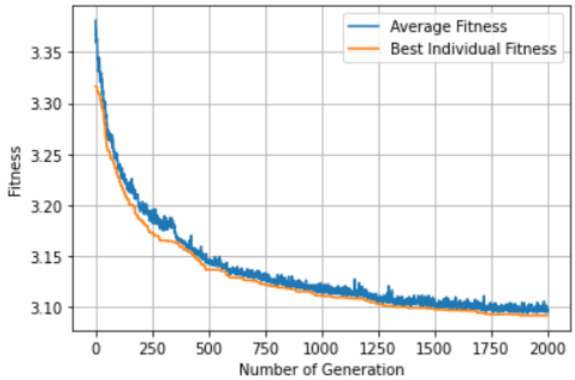
2.3 Genetic Algorithms

Evolutionary algorithms (EAs) are heuristic-based approaches, subset of computational intelligence, emanating from the Darwin's Evolutionary theory [7]. An EA implies processes inspired from the filed of biology.

A genetic algorithm (GA) is a type of evolutionary algorithm, part of the field of machine learning, mainly used for optimization problems. At the beginning of the process, there is an initial population, the generation g_i, regrouping elements called individuals. Each individual is assigned a fitness after an evaluation step, which corresponds to a score relating to the problem. Then, the selection step will keep the "best" individuals belonging to the current generation. Generally, we use a top-k selection where the k best individuals having a fitness closer to the optimal solution are picked. On those selected individuals, depending on the rates, there will be a chance that a mutation and a crossover occur. At the end, we obtain the new generation g_{i+1} of individuals, that, only guided by random events, can be better than the previous one : g_i. By repeating this process along N generations, the solution will continue to be as close as possible to the optimal solution. Figure 2.3 illustrates the process and the loop of a GA (2.3a) and how the fitness evolves over generations for a minimization problem (2.3b).



(a) General workflow of a genetic algorithm



(b) Example of a fitness evolution for a minimization problem after the execution of a genetic algorithm.

Figure 2.3: Genetic algorithm concepts and example of a fitness

Nevertheless, GAs implementation may be complex depending on the problem and they might be computationally expensive.

2.3.1 Optimized Pictogram Grid Generation using GAs

Recently, Carlos Vargas *et al.* [28] explored a first approach of automatic generation and optimization of a pictogram grid by applying a genetic algorithm for a minimization problem. The fitness of the GA corresponds to the grid cost from Lucie Chasseur *et al.* [5]. This work is preliminary and small scale, but leading to satisfying results, it represents an encouraging starting point for optimized pictogram grid generation.

2.3.2 Adaptive Genetic Algorithms

The difficulty of using a genetic algorithm is to find and calibrate parameters and operators correctly, like mutation rate or the number of generation, to solve the problem as much as possible. As a result, GAs may be slow to converge towards the solution or may be stuck in a local optimum. Recent works, show Adaptive Genetic Algorithms (AGAs), where parameters are evolving dynamically and automatically during the process [8]. AGAs by their adaptive approaches, lead to better performances and solve many of the GAs issues described above. Nowadays, AGAs are more attractive than GAs [12][15].

2.4 Natural Language Processing contributions for AAC

According to the definition from K.R Chowdhary [6], Natural language processing (NLP) is *"a collection of computational techniques for automatic analysis and representation of human languages, motivated by theory"*. Machines understand the meaning of a text to solve problems and perform different tasks such as word sense disambiguation, translation, text classification, etc. We list several NLP contributions for AAC and pictogram systems regrouping the main approaches and models.

2.4.1 Automatic pictogram generation from spontaneous speech

Most of the time, speech synthesis from pictograms can be part of different applications. However, the generation of pictograms starting from an audio signal such as speech is not common and much more complex. Previous works present an approach initially based on *Text2Picto* [27] using Automatic Speech Recognition (ASR), Text Simplification and Word Sense Disambiguation (WSD) [29]. The pipeline consists of four main steps. First, there is an ASR module (using an HNN-DNN model) that transcribes the input signal into text. This text is prepossessed using a lemmatizer and then simplified by a syntactical simplification or keeping only elements that are semantically important. At the end, from this lighter text, the system will retrieve pictograms associated to each words and build the corresponding sequence.

The *PRojection du langage Oral vers des unites PICTOgraphiques* (PROPICTO) project [31] aims to pursue this work and improve the speech to pictogram translation systems.

2.4.2 Predictive Models for AAC

The ability of predicting what the user wants to say may significantly improve system efficiency and communication speed. Originally, statistical language models were generally used for predictions (n-grams, etc.). With the arrival of deep learning, sequence to sequence neural network based models appeared in the last few years such as RNNs (Recurrent Neural Networks) [26] or BERT (Bidirectional Encoder Representations from Transformers) [9]. These end-to-end approaches (encoder-decoder) have been used many times for prediction tasks in NLP. We present a few examples of applications of those models for AAC.

2.4.2.1 Statistical Approaches and Models

For several years, statistical models had a central place in NLP field (machine translation, prediction, etc.). The user interface SIBYLLE helps people with impairment to enter text on a computer [30]. This system includes a stochastic language model that learns from user's texts and predicts dynamically the words that fit the most with the context. Based on a 4-gram, the model will estimate the probability a word occurs after a context composed of 3 words. In the end, it will display a list of candidate words to be selected by the user. The Brain-Computer Interface (BCI) helps the user by predicting the following potential letters using statistical models combined to electroencephalography (electrical activity of the brain) [19].

2.4.2.2 PictoBERT for Pictogram Prediction

Recently, a contribution presented a brand new model called PictoBERT, adapted from BERT and based on transformers, for a pictogram prediction task [20]. Those models are principally used for sequential treatment (text,sounds,etc.) in a self-supervised manner. PictoBERT will use the attention mechanism to find similarities and correspondences between embedded pictograms (tokens) from unlabelled data. Predicting the next pictogram that will be selected is an helpful support and it speed up the pictogram sequence construction.

— 3 —

Automatic Generation of an Optimized Pictogram Grid

3.1 Overview of the Optimization Pipeline

There are a few steps before obtaining an optimized pictogram grid starting from a raw corpus. We present the pictogram grid optimization pipeline we introduced and followed during this work. Figure 3.1 illustrates the overview of those different steps.

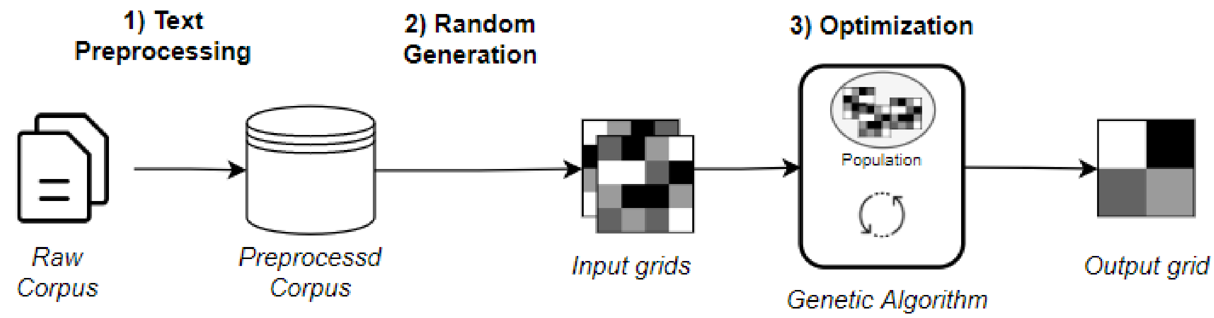


Figure 3.1: Steps of the optimization pipeline : 1) Text preprocessing to obtain a related corpus to the pictogram language. 2) Generate random grids with their associated tree-page structure. 3) Optimization process using a genetic algorithm.

First, we need to preprocess a raw corpus to make it fit with the language of pictograms. Then, we generate random grids from the preprocessed texts. This implies to correctly define a grid structure and make design choices. Once we have our initial population of random grids, the genetic algorithms will perform an optimization and outputs the best candidate.

This chapter presents the core of the optimization pipeline while the next chapter is about the dataset and the preprocessing step.

3.2 Pictogram Grid Structure

We describe a pictogram grid and how it is structured. First, we define all elements that constitute a grid. Then, we explain our choices about the tree structure representing the links between different pages and how we navigate within the grid. Appendix A.1 shows another structure and choices we initially explored.

3.2.1 Grid Hierarchy

The pictogram grid hierarchy is mainly inspired from previous works (Carlos *et al.*[28]) and existing applications such as AugCom [31].

3.2.1.1 Pictogram

The pictogram is the smallest component of a pictogram grid and also the most important one. In our work, a pictogram is only represented by a word and not an icon from any image banks. We do not require to associate icons to each pictograms during the generation process as we are only using words. Each pictogram also has the information of the page it belongs to and its position. A directory pictogram is a special pictogram pointing to another page.

3.2.1.2 Page

A page is a matrix filled by pictogram and it may contain empty slots or can be full. Each page has an unique name and cannot includes two identical pictograms. There is always a link between a directory pictogram and a page. The size of a page may be set before the generation, but will be fixed during the optimization process.

3.2.1.3 Grid

The pictogram grid is the final object, the one we want to automatically generate and optimize. A grid regroups several pages of pictograms and may includes duplicate pictograms (not in the same page). The first page of the grid is called the root page. The vocabulary of a grid is a set that contains distinct pictograms and excludes directory pictograms.

3.2.2 Tree Structure and Navigation within the Grid

In the previous section, we presented the grid hierarchy and defined each object. However, we did not discuss about the navigation and links between pages.

We maintain the page links by using a page-tree structure. Each node of the n-ary tree corresponds to a page. Obviously, the root page is the root of the tree. For instance, if a page P includes three directory pictograms, the node "P" will have three children. Figure 3.2 illustrates the two structures we maintain : the pictogram grid and the page-tree structure.

To navigate from a page to another, we just follow the tree links to go up or down (parent-child).

We adopted a tree structure, that will be useful for the evaluation complexity we will discuss later. However this structure prohibits two directory pictograms pointing on the same page as some pictogram grid applications do. Nonetheless, we can replace the tree structure by a graph to address the problem, but it might increase the complexity of the cost computation.

3.2.2.1 Page and Tree dimensions

The page dimension and the tree depth may have an important impact on the grid evaluation cost. Those parameters mainly depend on the movement cost and the selection cost related to the gird interaction method of the user (see Section 2.2.3 for more details). For instance, if

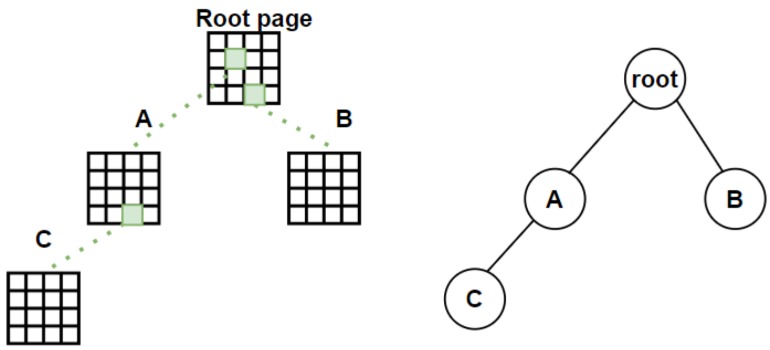


Figure 3.2: Example of a pictogram grid structure. On the left, the different pages with their directory pictograms leading to next pages. On the right, the associated page-tree of the grid.

the selection cost is high, we may reduce the number of pages and the depth of the tree and increase the page dimension to limit as much as possible the number of selections.

For now, we decided to let the user decide the pages dimensions (r,c) with r the number of rows and c the number of columns. Thus, the number of pages p for a total of n pictograms is $n/(r*c)$. As the page dimension is fixed, we want to reduce as much as possible the depth by a Breath First completion. For example, if we have 3x3 pages, the second level in the tree will contain 9 nodes and the third level will contain 81 nodes. The main drawback of this approach is that the user cannot know the best page dimension ahead of time.

Future researches could be done about a more adaptive method, finding the most suitable dimensions taking into account the number of pictograms and the different costs (movement and selection).

3.3 Evaluation of a Pictogram Grid

The crucial step in the optimization pipeline is the ability to automatically and rapidly evaluate the efficiency of a pictogram grid. However, as we have seen, because of the grid and display diversity, the grid evaluation remains challenging. The evaluation proposed by Lucie *et al.* [5] was a first exploration to compute a cost of a grid, but it did not take into account the different grid displays (see Section 2.2.2). Computing the distance between words of a sentence in the grid G is related to the syntax but not to an activity or a taxonomic display. We come up with a new evaluation keeping previous contributions and introducing a new evaluation approach considering multiple displays. The idea is to compute two different costs : the distance and the coherence. Furthermore, we inject a coherence coefficient α as a cursor to evaluate different display types. At the end, we obtain a new evaluation formula for pictogram grids :

$$Cost(G) = log(Cohe(G)) * \alpha + log(Dist(G,C)) * (1-\alpha) \tag{3.1}$$

where $Cost(G)$ is the cost of the grid G, α the coherence coefficient, $Cohe(G)$ is the coherence cost of the grid and $Dist(G,C)$ is the distance cost of G, using the corpus C. The more α is important, the more the coherence of the grid will be considered. If $\alpha = 0$, the evaluation will only take into account the distance between pictograms in a sequence and not the coherence of the pages. The computation of the two costs is independent and may be done in parallel.

3.3.1 Distance Cost

The distance cost is mostly inspired by recent works from Lucie *et al.*. [5] We want to evaluate the cost to build a pictogram sequence by computing distances between pictograms in the grid. A previous approach was transforming the grid into a graph where nodes are the pictograms and edges correspond to the distances between pictograms. The complexity was too high to compute this cost[1] for each new individuals of each steps of the genetic algorithm. Our approach is to use the tree structure of the page tree to find the shortest path by finding the Lowest Common Ancestor (LCA) between pages [1]. The number of selection will be the length l of this path. We considerably reduce the number of nodes as those are pages and not pictograms. We fist compute and store an Euler Tour of the tree before the evaluation. This helps to find the LCA in the page tree with a complexity of $O(1)$ and then the shortest path in $O(depth)$. We also kept the Movement Cost C_M and the Selection Cost C_S that vary according to the navigation method (eye track or touch).

In the end, we obtain the distance cost $Dist(G,C)$ as :

$$Dist(G,C) = \sum_{i=0}^{n} \sum_{j=0}^{m} (C_M * d + C_S * (l+1)) \tag{3.2}$$

where for a given grid G and a corpus C, for n lines in C, we sum the m distances between two pictograms. We multiply d, the total euclidean distance from every pages we cross to the Movement Cost C_M and l the shortest path length to the selection cost C_S.

The total euclidean distance d is computed as :

$$d = \sum_{i=0}^{l} dist(p_{start}, p_{target}) \tag{3.3}$$

with p_{start} and p_{target}, the two pictograms of the page i for which we will compute an euclidean distance and l corresponds to the length of the path, so the number of pages and euclidean distance we will compute.

For an entire pictogram sequence, we might have different paths. We make the choice to apply a greedy algorithm, selecting the shortest path between two pictograms each time, instead of the whole path. Obviously, it is less optimal but it implies a lower complexity and is closer to reality (the user will not predict the optimal path before building its sequence, but may think of the best path for the next pictogram).

3.3.2 Coherence Cost

As we have seen before, taxonomic or activity displays regroup pictograms that are semantically similar. Our goal is to use a language model such as Word2Vec [16] to encode words into vectors and compute the cosine similarity between them. We chose the fastText model [3] because of the language diversity it proposes (e.g : english, french, ...) and the infinite number of vector formation by using subwords. The idea is to compute the coherence Cp_k of each page by adding cosine similarities between each word w_i to all words of the page of size Sk. However, we want to solve a minimization problem. To do so, we do not want word similarities but we

[1]For n pictograms and one page in the grid, the worse case complexity is $O(n^2)$ to build the graph.

compute and try to minimize word dissimilarities. In the end, we sum up the page coherence's of the K pages. The coherence cost $Cohe(G)$ is defined as :

$$Cohe(G) = \sum_{k=0}^{K} \sum_{i=0}^{S_k} \sum_{j=0}^{S_k} (1 - cosine(\vec{w}_i, \vec{w}_j)) \tag{3.4}$$

where K is the page number, Sk is the size of the page and $\vec{w}_i, \vec{w}_j$ are respectively the associated vectors for the words w_i, w_j.

3.4 Genetic Algorithm for Pictogram Grid Optimization

As the preliminary version from Carlos *et Al.* [28] was a significant start and brought promising results to automatically generate an optimized pictogram grid, the genetic algorithm stays at the heart of our pipeline. Next sections describe our parameters and implementation choices.

3.4.1 Population and Parameters

Undeniably, an individual in our GA corresponds to a pictogram grid. The initial population consists of N randomized pictogram grids. Each grid includes the same vocabulary extracted from a corpus. The process will try to minimize the grid fitness $Cost(G)$ along the g generations. During the optimization, each genetic operator depends on parameters such as the mutation probability or the selection number. Finally, the evaluation step will mainly depend on the coherence coefficient α we already discussed in section 3.3. In a first place, all of those parameters will be configured by the user and might have an important impact on the optimization.

3.4.2 Genetic Operations

Genetic algorithms are not universal, they must be adapted to the problem. A common simple problem using GAs involves binary encoded chromosomes with elementary operations [13]. However, in our case, crossover and mutations should be more elaborated than binary operations. Thus, it is necessary to properly define each step of the process. We introduce genetic operators (crossover, mutations) related to the pictogram grids (individuals).

3.4.2.1 Selection

For the selection step, we pick the individuals that are closer to the optimal solution out of the population. In our case, we decided to perform a top-k selection. As we want to solve a minimization problem, we select the k best individuals having the lowest fitness within the whole population. The selection number (k) will be decided by the user for the optimizer configuration.

3.4.2.2 Crossover

The crossover is the step that forms a new individual by leveraging information from the parents. The main point is not to have vocabulary information loss after the operation. We come up with a pictogram position-based crossover operation. During the process, the apparition of a crossover depends on the crossover probability P_{cro} . Figure 3.3 illustrates the crossover process between individuals X and Y, engendering the new individual X'.

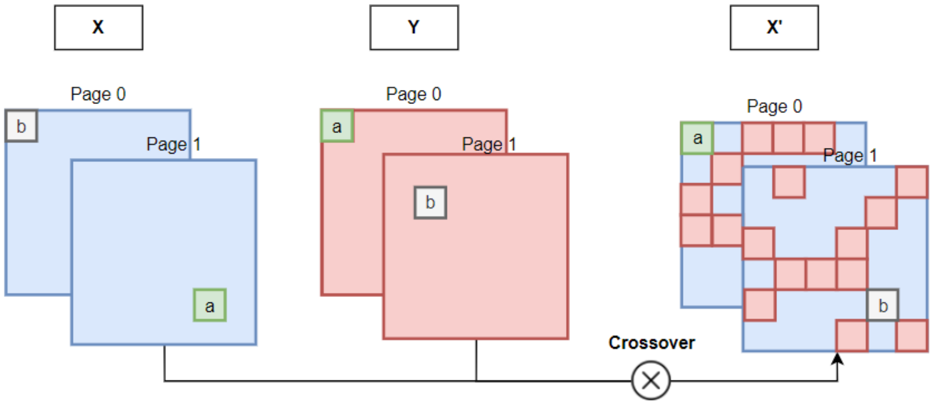


Figure 3.3: Pictogram position-swap-based crossover operation. Given a CIR = 0.5, the half of the pictograms position information from the Y individual is transmitted to X', a copy of X

Given a Crossover Information Rate (CIR) and the two parents X and Y having the same vocabulary, the crossover operator selects from Y, a proportion of pictograms depending on the CIR, to tell their position to X'. In the end, the offspring X' is a copy of X modified by the position information from the selected pictograms of Y [2].

3.4.2.3 Mutations

As crossover, an individual must mutate without loosing any vocabulary information. The mutation can take place according to the mutation probability P_{mut} . We introduce four new appropriate mutations for pictogram grids. Figure 3.4 illustrates the four mutations that can happen during the optimization process.

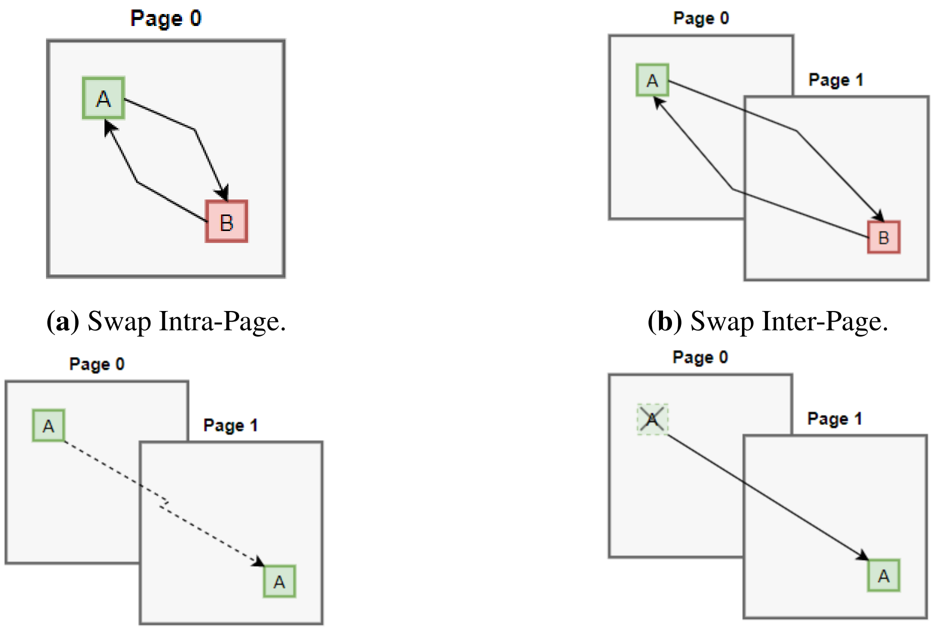
(a) **Intra-Page Pictogram Swap**

The idea is to swap the position of two pictograms belonging to the same page. The pictogram directories can also be moved as the children's links will not be lost. Obviously, this mutation will only impact the final distance cost and not coherence (as we sum the page similarities).

(b) **Inter-Page Pictogram Swap**

Unlike the swap intra-page, the swap inter-page allows to exchange the position of two pictograms coming from two different pages. We impose to only swap directory pictograms if they have the same depth. The main reason is, in a first place, not to implement a complex mutation that can modify the structure of the tree and its depth. This mutation may have an impact on the coherence cost.

[2] A part of the position information given from X will be lost as we will swap the pictogram holding the target position with the selected one.



(a) Swap Intra-Page. **(b)** Swap Inter-Page.

(c) Cloning **(d)** Displacement

Figure 3.4: Different mutations operators : a) Two pictograms from the same page exchange their postions. b) Two pictograms from two different pages exchange their position. c) A from one page is cloned into another page. d) A pictogram is displaced from a page to another.

(c) **Pictogram Cloning**

Previous mutations involve only swaps between pictograms. The idea behind pictogram cloning is to copy a pictogram and to append the clone in a non-empty page. This allows the grid to have duplicates within different pages. However, this mutation will increase the number of pictograms in the grid and make it heavier.

(d) **Pictogram Displacement**

The pictogram displacement will remove one random pictogram from a page and add it to another empty page that does not include it. Pictogram directories are not affected by this mutation.

For now, each mutation occurs randomly with a fixed probability P_{mi} . However, depending on the situation (configuration, parameters, initial grid, etc.), some mutations may have less impact on the fitness than others. We planned a possible adaptive approach for mutations to apply on our genetic algorithm. The idea is to set an initial probability for each mutation to occur, and at each generation, update each probability depending on the effect of the mutation on the fitness, following a gradient descent. Appendix A.2 presents our adaptive mutation suggestion in more details.

3.4.3 Multiprocessing and Parallelization

As already discussed in previous sections about genetic algorithms, we want to explore as much as possible the solution space and not fall and stay in a local minimum. A possible approach to solve this issue is to use multiprocessing. We assign an optimizer to each process with different configurations and parameters. The more core we have, the more we can cover the solution space. Thus, there will be simultaneous optimization and at the end of the pipeline, we keep the best candidate from all the processes.

The genetic algorithm execution may be slow and we plan to later reduce this time by taking advantage of GPUs for repetitive tasks such as genetic operations or the grid evaluation cost computation. Indeed, for the same generation, the workflow is the same for each individual of the algorithm some operations are independent.

3.5 PictoGriz : a Pictogram Grid Visualization Tool

In parallel of our work, we developed a visualization tool called PictoGriz. The main purpose is to visualize in an efficient way the results of the automatic generation of optimized pictogram grids and navigate through them.

3.5.1 Overview of the tool

Our purpose is to observe the whole grid at once. As the grid structure is levelled (tree), we decide to remove the tree dimension by arranging pages side by side. In the end, the final grid looks like a huge matrix containing squares representing every pictograms. Figure 3.5 is a screenshot of the tool and shows how it is organized. For future version, we should separate the different pages with spaces and draw the links between directory pictograms and their corresponding page. The different interactions aim to help the navigation and the information research in the grid.

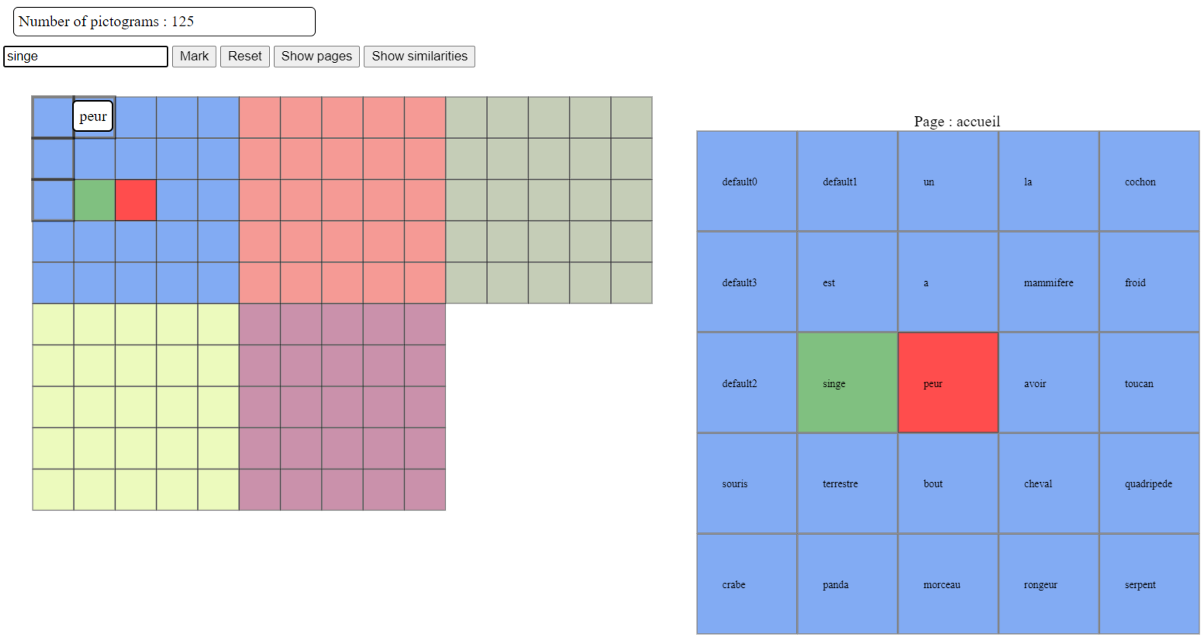


Figure 3.5: Screenshot of the global view of the grid visualization tool. The top-left part corresponds to the information of the grid and possible interactions (Search, Mark, Reset, Show Pages, Show Similarities). The square grid of the left represents the flat vision of the pictogram grid. The right part is a zoom on the selected page to display words.

3.5.2 Implementation and Details

This tool is mainly written in JavaScript and HTML. It is using the D3js library which is an helpful library for managing data and visualization [4]. There are about a total of 500 lines of code. The tool is loading a pictogram grid in a Comma-Separated Values (CSV) file (following the AugCom format) and converts each pictogram in a small square. The directory pictograms have a wider border to differentiate them from the others. The page colors are randomly generated. We can differentiate the different pages, but if they are too many, the color number will be too high and the colors might be confusing.

3.5.3 Interactions

The embed interactions are crucial when using the tool. First, the research bar allows the user to search a word. As grids may involve large-sized vocabularies, it should be a considerable loss of time to search words by hand. Once the word is found, we can mark it thanks to the "Mark" button to save its location when we will change the word in the research bar. The two other buttons are in order, "Show pages", applying colors to differentiate pages and "Show similarities" to apply a gray gradient on each pictogram depending on their similarities (the higher the similarity is, the darker the pictogram is). Furthermore, we can navigate within the grid thanks to the mouse displaying a zooming effect that shows each pictogram belonging to the same page.

3.5.4 Future Improvements

This tool is a first contribution to rapidly navigate through a pictogram grid and analyze it. For now, this visualization satisfies our expectations, and we used it frequently. For later on, we plan to include different features to help the user even more. First, we plan to display the links between different pages to be aware of the page-tree structure and pages dependencies. Another suggestion is to insert a distance cost calculator in real time. The purpose of this module is to allow the user to compute the distance cost between two pictograms within the grid. Finally, we should increase the visibility of the tool such as page boundaries, color codes, font-size in the zoom window, etc.

— 4 —
Dataset

As we presented in the optimization pipeline, we require a dataset and perform a preprocessing on it to fit with the usage of pictograms. In the following sections, we will develop our vision about pictograms datasets and we will present the choices and work we done about text corpora.

4.1 Dataset and Preprocessing

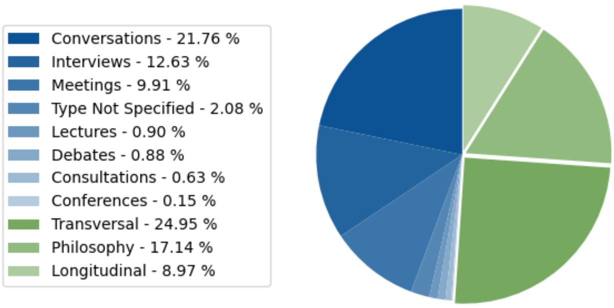
Two categories of datasets are useful for pictogram processing. Pictogram datasets are image banks gathering pictogram icons with corresponding labels. Generally, softwares or applications make use of those datasets for the benefit of users, but they are less interesting for NLP purposes. The other category regroups the text or speech datasets including annotated (or not) corpora. What is surprising is we do not work with images for pictograms but with words instead. Because of the variety of meanings a pictogram may have corpora are mostly word-sense annotated. The word-sense disambiguation task (WSD) is the attribution of the correct meaning of a word given a context. For instance, a WSD model is able to differentiate *Fly (the insect)* and *Fly (moving through the air)* from the sentence. The idea, by working with pictograms and WSD, is for the pictogram to make sense with regards the context. WordNet is one of the most popular word-sense annotated database, regrouping words into set of synonyms (synsets) [17]. Previously cited contributions like the Automatic Pictogram Generation from Speech [29], make use of WordNet, where Jayr Pereira *et al.* [20] built the SemCHILDES dataset as WordNet is too small for PictoBERT training.

4.1.1 TCOF Dataset

In a first place, we do not necessarily need WSD for optimized pictogram grid generation as the distance cost for the evaluation, mainly depends on the sentence syntax. Also, as we collaborate with french speech-therapists, we selected a french corpus for our experiments.
The *Traitement de Corpus Oraux en Français* (TCOF) dataset, from Virginie André *et al.*, provides speech and associated transcriptions from Adult-Child or Adult-Adult interactions [11] [33]. Texts or speech from TCOF are not word-sense annotated. The main reason of this choice of dataset is the closeness of the Adult-Child interactions with the AAC domain and the concrete use of pictogram grids. Generally, people will not employ too complex or too variate vocabulary similarly to pictogram sequences formation (even though, larger and more

complex grids exist and can be used for AAC purposes). Furthermore, as there are speeches and transcripts, it lets the choice to process texts or perform ASR.

The first version of the TCOF regroups about 517 transcriptions from 134 hours of speech in total. As Adult-Child interactions are more interesting for our work than Adult-Adult, we decided to focus on them and, first, on the transcriptions. The Adult-Child section is divided in three parts : Longitudinal Corpus, Transversal Corpus and philosophy interviews. The raw transcriptions contains some symbols marking the pauses or misunderstandings from the transcripter. Figure 4.1 shows the distribution of the different categories of speech we can find in TCOF. The Adult-Child section represents more than half of the dataset with a duration of 77 hours and more than 700 000 transcripted words. [1].



(a)

Adults-Adults			
Adult parts	Speaker changes	Words	Duration (h)
Conferences	1	2 169	0.16
Consultations	1 033	9 020	1
Conversations	23 084	311 084	24
Lectures	443	12 899	1.8
Debates	673	12 613	1
Interviews	9 722	180 487	15.5
Type Not Spec.	333	29 721	2.7
Meetings	11 797	141 602	11
Total	**47 086**	**699 595**	**57.16**

Adults-Child			
Child parts	Speaker changes	Words	Duration (h)
Longitudinal	14 331	128 210	15.3
Transversal	51 600	356 594	43.3
Philosophy	30 060	244 981	19
Total	**95 991**	**729 785**	**77.6**

(b)

Figure 4.1: TCOF Dataset Statistics : a) Distribution of the different categories of speech. b) Table of the size of each categories of speech (speaker changes, words, duration).

4.1.2 Preprocessing and Vocabulary

The transcriptions from the TCOF dataset are complete sentences, which are commonly used in the french language. However, this is far from the pictogram communication as all the words are not encoded into a drawing. Furthermore, those raw transcriptions are not usable because of the transcripter symbols and the missunderstandings. Our main objective is to extract a simplified and smallest vocabulary from the corpus to be as close as possible to the language of pictograms. We also need to clean the texts (noise, transcripter symbols, etc.).

4.1.2.1 Pipeline for Corpus Preprocessing

The preprocessing of a non word-sense annotated dataset consists in two major steps : a lemmatization and a suppresion of unwanted words. Figure 4.2 illustrates the preprocessing pipeline of the TCOF transcriptions. Each sentences of the corpus will have the same treatment. First, we want to apply a lemmatization using a language model to only get lemmas. Then, we filter

[1]Some words or sentences may be noisy or correspond to transcripter symbols such as '***'.

the stopwords and non existing words thanks to a dictionary. In the end, we obtain sentences that are very similar to pictogram sequences.

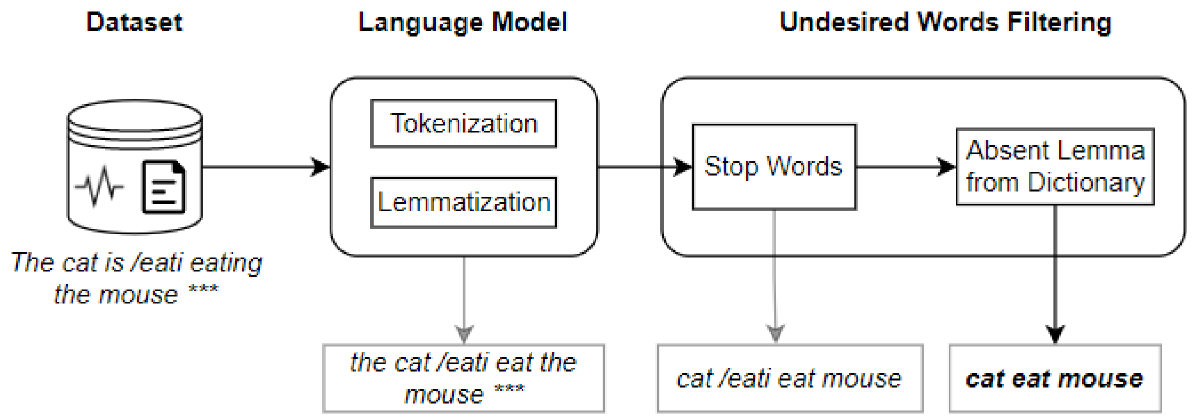


Figure 4.2: TCOF Preprocessing Pipeline from transcriptions to pictogram language. First, lemmatization of the sentences with a language model. Then filtering of the undesired/noisy words.

4.1.2.2 Language Model for Lemmatization

The first step of the preprocessing pipeline is to use a language model to tokenize words and apply operations such as lemmatization, part of speech tagging, etc. The Stanza toolkit provides neural NLP pipelines for many languages including English or French [22]. In our case, we use the *French GSD* model offered by Stanza. The word tokenization is the first step of an NLP pipeline. The idea is to split the sentences into tokens (each word will have an associated token). Those tokens are useful for the continuation of the NLP pipeline. After tokenization, the most important task we need is the lemmatization. The model will retrieve the associated lemma of each input token. A lemma is the basic form of a word that we find in dictionaries. On the example from the Figure 3.2, *eat* is the lemma of the following set of tokens [*eats, eating, eat, ate*]. Lemmas are crucial for pictogram grid generation as we want to limit the number of different pictograms as possible.

4.1.2.3 Undesired Words Filtering

After lemmatization, the sentences contain mostly lemmas but also undesired words. There are two types of words we do not want to keep in the texts. First, the stopwords are words we will filter because of their syntactic function. Our objective is to be as close as possible to the pictogram sequences and words such as determinants, punctuation marks or onomatopoeia might not be appropriate. We built a list of stopwords and we remove each token belonging to it. The other category is the lemma that are not belonging to the dictionary. Speeches or transcriptions cannot be perfect and may contain some noises (missunderstandings, bad pronunciations from the speakers, etc.). For instance, in the TCOF texts, there are transcripters symbols or word fragments that are not part of the French language. Obviously, for pictograms or any usages, it is preferable not to have non-existing words. To remove those undesired words, we check if the tokens belong to a dictionary. DBNary, from G. Serasset *et al.*, is a lexical resource, extracted from the Wiktionary providing information about words such as the definition, the sense, etc. [25]. We built our french dictionary by only extracting the words from DBNary. In the end, our dictionary will contain almost all the french existing words and we will use them to filter our sentences. Figure 4.2 sentence example contains stopwords such as *the* and lemma from transcripter symbols that are absent from DBNary like [*/eati,* ***].

4.1.2.4 Adult-Child TCOF Results

We present the results obtained after the preprocessing step on the TCOF dataset. The two purposes of the preprocessing is the pictogram vocabulary extraction and to make appropriate sentences. As said previously, we only operated on the Adult-Child section of the TCOF as it represents almost 60 hours of transcriptions and as the Adult-Adult part is less relevant. First, the final vocabulary regroups 4 800 different french words (nouns, verbs, etc.). For now, there are still some noisy words we did not catch [2], but we can add them to the list of stopwords and filter them in the future, following an iterative process. Then, the final sentences of the corpus are much lighter and similar to pictogram sequences. In the end, we reduce the total number of words of the Adult-Child section from about 700 000 to 206 460 ($\sim 71\%$ of words removed). Some preprocessed sentences are presented in Table 4.1. The first sample reflects the idea of simplification of the sentences to be similar to a pictogram succession. For the second one, the preprocessing also removed the transcripter symbols thanks to DBNary. However, the main drawback is the loss of context because of the list of stopwords. As we can observe in the second output sample, the sentence does not include *"elle"* anymore, so we loose some information about the context. We also get rid of the negation marks such as in the third sentence, the word *"pas"* is filtered using the list of stopwords. Proper nouns such as *Alban* in the last sample, are also removed as it is close to impossible to draw a pictogram for each name.

Input sentence	**Output sentence**
les abeilles et les moustiques qui piquent ?	*abeille moustique piquer*
elle descend du camion /// tu arrives à remettre ? ///	*descendre camion arriver remettre*
non parce qu'ils pensent pas vraiment +	*non penser vraiment*
tu m'as dit que Alban il t'avait donné un coup de pied que c'était plus ton copain	*donner coup pied copain*

Table 4.1: Samples of preprocessed sentences from the TCOF dataset.

Our choices are rather restrictive to simplify the sentences but may impact the meaning. To conclude about the preprocessing step, results mainly depends on the way we filter the words (list of stopwords, DBNary, etc.) and the transcriptions' quality. The user will apply restrictions to target the final vocabulary and therefore what the sentences will contain.

[2] The filter depends of the user choice. We could suppress more onomatopeia, single letters, proper nouns, ...

— 5 —
Experiments and Evaluation

This chapter presents the different experiments we have done to assess our approaches. First, we test different grid generations depending on the coherence coefficient α. Then we conduct a task evaluation to compare pictogram grids efficiency with existing standard grids such as PODD.

5.1 Automatic Optimized Pictogram Grid Generation

We want to test the generation of optimized pictogram grid by using the genetic algorithm we presented in section 3.4. We generate different grids to test the influence of the α coefficient and the different pictogram grid displays we can obtain. We employ our pipeline (see section 3.1) on different corpora to have small-scale grids and larger grids.

5.1.1 Configurations and Parameters

To be able to compare generation results, configuration of the process has to be the same for each type of grids we optimize. As we use multiprocessing, there will be four different configurations (for each process) to offer a diversity of genetic parameters and further explore the solution space. Table 5.1 describes the configuration processes.

	Genetic Operators			
Processes	P_{cro}	CIR	P_{mut}	Selection Number
P_0	0.5	0.5	0.5	30
P_1	0.7	0.4	0.3	40
P_2	0.3	0.8	0.4	30
P_3	0.3	0.6	0.7	40

Table 5.1: Configuration processes of the genetic operations. Parameters : P_{cro} the Crossover Probability, CIR the Crossover Information Rate and P_{mut} the Mutation Probability.

We thought about four configurations to test as much as possible the different possibilities. The process P_0 is the more balanced one, with probabilities and a CIR of 0.5. P_1 and P_3 are the opposites, with P_1 we want to test the efficiency with more crossover (P_{cro} = 0.7) than mutation (P_{mut} = 0.3) while with P_3 there will be much more mutations. P_2 has poor probabilities but a high CIR.

Other parameters are the same for all processes. Every page of the grid will have a dimension of 5x5. The initial grid population is set to 100 and the number of generation is 500 for one optimization[1]. The parameter α will be the same for each process, but it will differ depending on the generation test. Following parts show results about the different optimized grid generation we performed.

5.1.2 Coherence Coefficient Variation and Displays

As mentioned in Chapter 2 (sections 2.2.2 and 2.2.3), the pictogram grid displays are diverse depending on the situation and the usage. We test the new pictogram grid evaluation we introduced in section 3.3 with different values for α. The execution times for all the optimizations are presented in the Appendix A.4.

5.1.2.1 Text Corpora

For this experiment, we prepared two different text corpora, that will be the same for the three generation types.

First, we made up a very small french corpus that we called *AnimalTexts* (AT), to show the efficiency of our system on small texts. It regroups 99 unique words for 312 in total and 49 simple sentences about animals and fauna with an average length of 6.4 words. However, we did not preprocess it like we wanted to in order to really show the syntactic impact on the grid. Appendix A.3 regroups examples of sentences from the AT corpus.

The second corpus, is the preprocessed Philosophy part from the Adult-Child section of the TCOF dataset (see Chapter 4 and section 4.1.1 for more details). It represents 17% of the whole dataset with a vocabulary of 2 402 unique words and 9 635 sentences (60 151 words in total) of philosophical interviews with children aged 6 to 10 years old. For each test, we will generate and optimize one grid from the AT corpus and one from the Philosophy TCOF part.

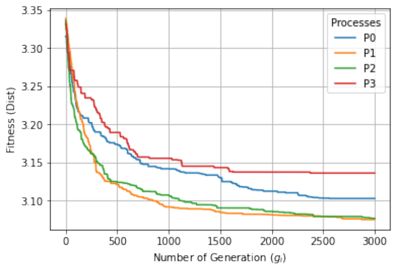
5.1.2.2 Distance Grids

By setting α to 0, we want to generate a grid that only depends on the syntax and the corpus. The fitness corresponds to the distance cost $Dist(G)$ (see section 3.3.1) and the coherence is not considered. We target a semantic-syntactic display, so we expect proximity between words following the syntax and generally used one after the other.

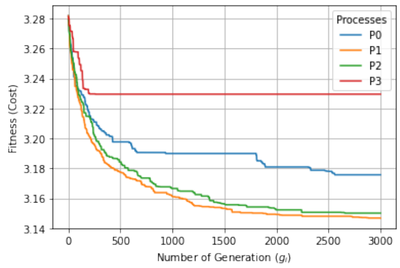
First, we generate the distance grid AT_{Dist} from the AT corpus. Figure 5.1a represents its corresponding fitness evolution during the optimization. We observe convergence differences among the processes, but they seem to stabilize after 1 500 generations. P_1 and P_2 have a lower fitness than the two other processes showing that crossovers might have a more significant impact than mutations. Figure 5.2a represents the content of the AT_{Dist} "accueil" page. We observe word clusters formed around one specific word (e.g : *"lionne, souris, girafe, tortue"* close to *"une"*). The reason is that words that occur the most in the corpus tend to be in the center or in the top-left of the page to considerably limit the euclidean distance.

Next, from the Philosophy part of the TCOF dataset, we optimize the distance grid Tcof_{Dist}. From Figure 5.6a, we observe that the stabilization starts at about 8 000 generations. Even

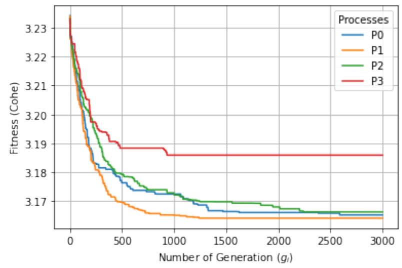
[1] The best grid is saved after 500 generation, but for a larger test, the process is iterated few times



(a) $\text{AT}_{Dist}(\alpha = 0)$ fitness. The only cost considered is $Dist(G)$.

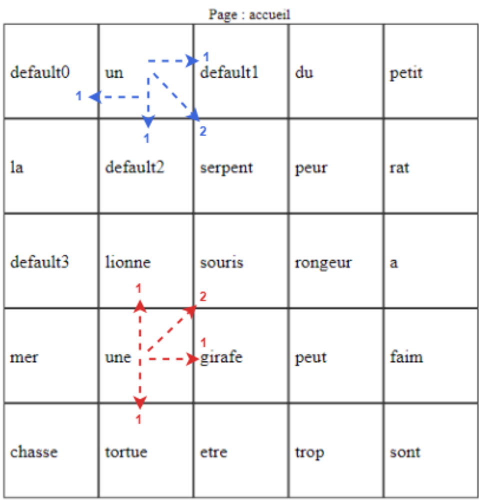


(b) $\text{AT}_{Hybrid}(\alpha = 0.5)$ fitness. Both $Cohe(G)$ and $Dist(G)$ impact the cost.

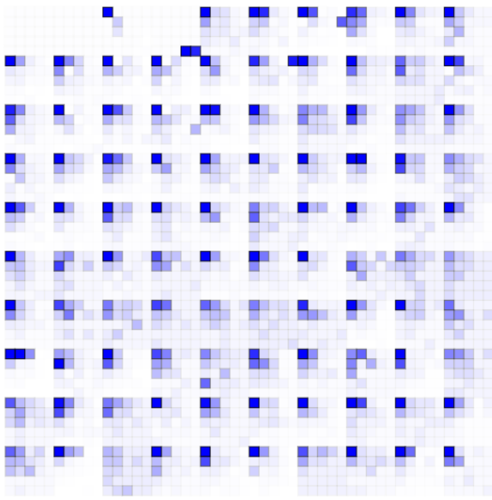


(c) $\text{AT}_{Cohe}(\alpha = 1)$ fitness. The only cost considered is $Cohe(G)$.

Figure 5.1: Fitness evolution of the different pictogram grid optimizations from the Animal-texts corpus. The three grid fitness converge well before the end of the execution. There are visible differences between processes, because the optimization



(a) Page "accueil" from AT_{Dist}. French feminine words are around and near to "une", while "un" is the center of masculine words.



(b) Heatmap representing word occurrences per page (5x5) of the Tcof_{Dist} pages. The color opacity is related to the word occurrences. For each page, top-left word is darker than the other.

Figure 5.2: Generation results for distance grids AT_{Dist} and Tcof_{Dist} : a) An example of a page from AT_{Dist}. b) The global occurrence heatmap of Tcof_{Dist}. The "defaultX" pictograms are the directory pictogram leading to the page "defaultX".

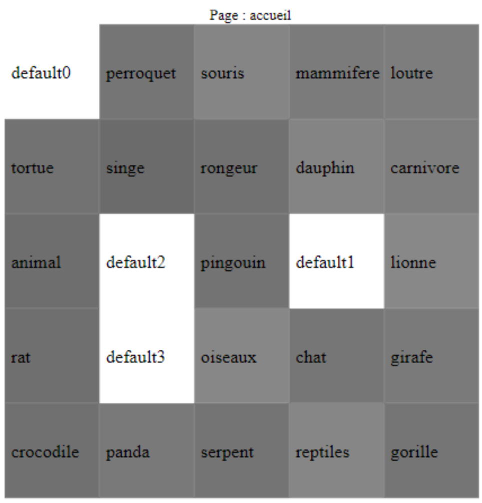
after 16 000 generations, as there are more words to handle and longer texts, the fitness still decreases. However, in comparison to previous AT_{Dist} the execution time is significant (see Appendix A.4 for more information about execution times). Figure 5.2b corresponds to the heatmap of the word occurrences for each page and confirms that the more a word is frequent, the more it will be placed on the top-left of the page.

5.1.2.3 Coherent Grids

To obtain a fully coherent pictogram grid, corresponding to a semantic display such as Taxonomic, we set the coherence coefficient α to 1. As the evaluation will only leverage coherence cost $Cohe(G)$ (see section 3.3.2), the fitness will depend on it and the final grid will be influenced by the semantic of the words. In the end, we should obtain a grid with pages containing words that are semantically close.

As AT_{Dist} the optimization of AT_{Cohe} (from AnimalTexts), figure 5.1c shows a rather fast convergence. This time, only P_3 differs from the other processes, because it performs less crossovers. Figure 5.3a represents the page "accueil" of AT_{Cohe}. This result is exactly what we expected, a page full of animals, reflecting the targeted Taxonomic display.

Next, we generate and optimize the $Tcof_{Cohe}$ grid. The example of a page from $Tcof_{Cohe}$ is presented in Figure 5.4a. We identify different semantic sets from the 25 words. For instance, the words *"miauler, tout-petit, ronronner, petit, mignon"* are related to *"chat"*. However, there is no unanimous category like we obtained with AT_{Cohe}. On Figure 5.6c, we observe that after 16 000 generations, the slope of the fitness is still steep, meaning the optimization is not finished. For preferable comparison, we decided to stop it to have an equal generation number between each grids. By repeating the process until convergence, $Tcof_{Cohe}$ pages should have a higher coherence.



(a) Similarity heatmap of the page "accueil" from AT_{Cohe}. Words only refer to animals. The global similarity is high.

(b) Similarity heatmap of the page "accueil" from AT_{Hybrid}. Words refer to animals but also to determinants used together. The global similarity is decreased because of the determinants.

Figure 5.3: Similarity heatmaps from pictogram grids optimized on AnimalTexts corpus : a) Page "accueil" from AT_{Cohe}. b) Page "accueil" from AT_{Hybrid}. Most related words to others appear in darker color.

5.1.2.4 Hybrid Grids

The two previous generation fitness followed either the coherence cost or the distance cost. Now, we set α to 0.5 to test an hybrid optimization and minimize the cost $Cost(G)$ (see section

(a) Similarity heatmap of the page "default35" from Tcof$_{Cohe}$. The words related to "chat" are highlighted in green.

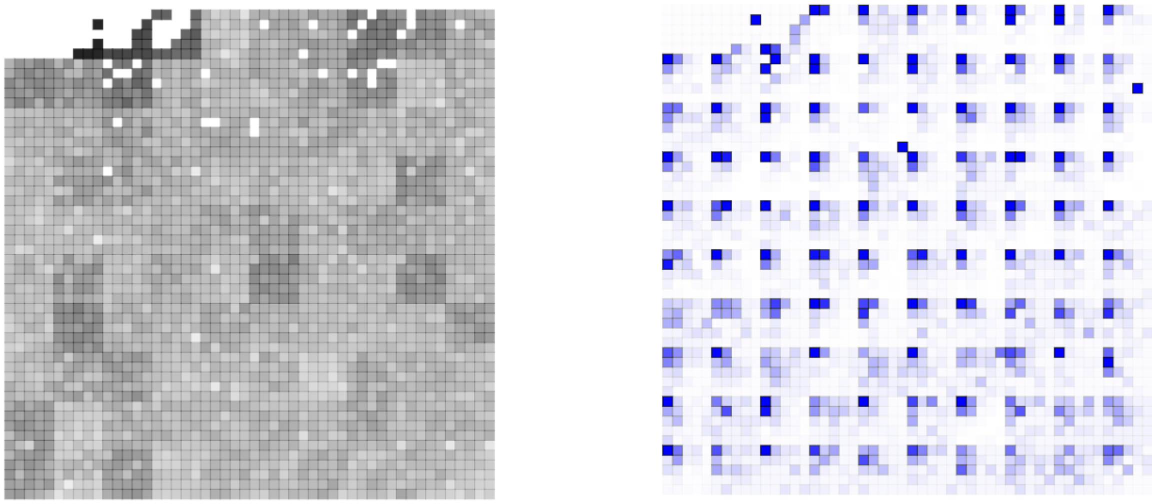
(b) Similarity heatmap of the "default69" page from Tcof$_{Hybrid}$. The words related to the "forest" thematic are highlighted in green.

Figure 5.4: Similarity heatmaps from pictogram grids optimized on TCOF Philosophy corpus : a) Page "default35" from Tcof$_{Cohe}$. b) Page "default69" from Tcof$_{Hybrid}$.

3.3) of the grids. The expected grid display should be similar to an activity/contextual display.

First, we optimize the AT$_{Hybrid}$ grid (appendix A.6 regroups the corresponding page-tree and all pages of AT$_{Hybrid}$). The fitness convergence is very similar to the AT$_{Dist}$ fitness as showed in figure 5.1b. Nonetheless, the content of the page *"accueil"* seems to be a combination of AT$_{Dist}$ and AT$_{Cohe}$. Figure 5.3b shows a page that is not full of animals, with on the top-left, the determinants appearing frequently and before a word that refers to an animal. The directory pictograms ("defaultX") are also close to the top-left to reduce euclidean distances when navigating in the tree.

Then, Tcof$_{Hybrid}$ is obtained from the TCOF Philosophy part. As for Tcof$_{Cohe}$, figure 5.6b shows the fitness does not converge totally, but the slope is less steep. The behavior of its fitness is between Tcof$_{Dist}$ and Tcof$_{Cohe}$ fitnesses. However, such as Tcof$_{Cohe}$, the coherence of the different pages is not fully satisfying. Figure 5.4b represents the "default69" page of the Tcof$_{Hybrid}$. We observe some semantic sets such as animals or trees, but a "forest trend" also seems to emerge (*"ortie, chêne, pommier, chevreuil, renard, abeille, loup-garou, pie, sanglier"*). Nevertheless, there are quite a few noisy words in the page. Again, this can be explained by the lack of iteration during the process, but also by the "hybrid" behavior that can suppress semantic information to satisfy the syntactic or the opposite. To get and analyze a global view of Tcof$_{Hybrid}$, Figure 5.5a represents the similarity heatmap of all pages and shows the impact of the coherence cost over the grid. We can identify the different pages, only by looking at the similarity scores. Also, we observe several pages with an important coherence but also with a low coherence. The reason of this contrast comes from random mutations that will move high-similarity words with low-similarity ones to increase the global coherence. In the end, undesirable words will be together as they do not increase similarity of other pages. Appendix A.5 provides page examples from the Tcof$_{Hybrid}$ to illustrate this phenomenon. Figure 5.5b, shows the distance cost's impact. Similarly to Tcof$_{Dist}$, the words have appeared frequently



(a) Similarity heatmap (per page) of the Tcof$_{Hybrid}$ grid. Pages emerged from the similarity. Darker pages contain semantically similar pictograms where lighter pages have a lower coherence.

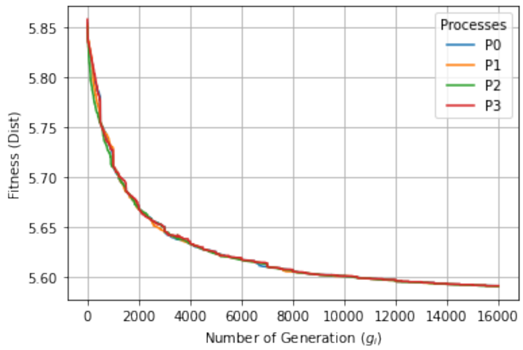
(b) Heatmap representing word occurrences per page (5x5) of the Tcof$_{Hybrid}$ pages. For each page, top-left word is darker than the other. However, there is less constrast than Tcof$_{Dist}$.

Figure 5.5: Overall analysis of the Tcof$_{Hybrid}$ grid : a) Corresponding similarity heatmap of every page. b) Corresponding occurrence heatmap of every page.

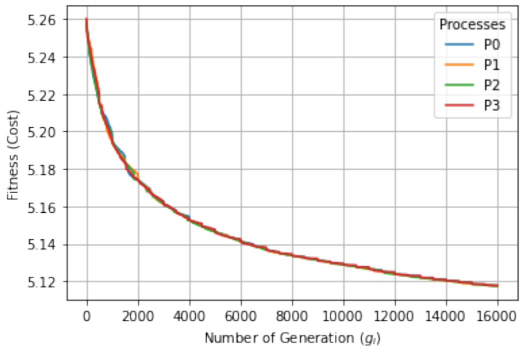
are in the top-left of each pages. In the end, by looking at those two heatmaps, we can say Tcof$_{Hybrid}$ has been influenced either by the coherence cost and the distance cost and is hybrid.

However, we do not obtain exactly the targeted contextual display for each pages. The example of the page from the figure 5.4b slightly matches with a contextual display, because of its *"forest"* thematic, but there are no verbs or adjectives to help the user form pictogram sequences and the presence of noisy words is not negligible. There are many reasons why we do not obtain contextual pages. First, we can iterate the process until convergence. Then, the language model we use tends to regroup words by their meanings but also by their Part of Speech (POS). For instance, in our grid, the majority of verbs are together. Also, we do not take into account WSD for the distance cost, which might impact the displacement of polysemic words. Another point to take into consideration is the quality of our preprocessed corpus. Maybe, our preprocessing choices (filters) led to bad quality sentences, loosing important syntactic information. Finally, genetic parameters, coherence coefficient (α) and cost computation may also influence the divergence from a contextual display.

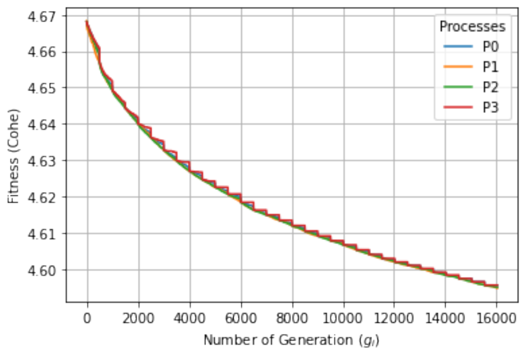
For our future experiments, we should vary parameters such as grid information (page dimension), genetic parameters and the coherence coefficient α to better measure its impact. We should also optimize grids, until convergence, from different sized datasets.



(a) Tcof$_{Dist}(\alpha = 0)$ fitness. $Dist(G)$ tends to converge after 10 000 generations.



(b) Tcof$_{Hybrid}(\alpha = 0.5)$ fitness. $Cost(G)$ seems to converge slightly, but more generations should be required.



(c) Tcof$_{Cohe}(\alpha = 1)$ fitness. $Cohe(G)$ does not converge yet, even after 16 000 generations.

Figure 5.6: Fitness evolution of the different pictogram grid optimization from the TCOF Philosophy part of the TCOF dataset. The distance convergence is faster than the coherence or the hybrid convergence. For each optimization, there are differences between the four processes after 500 generations, but not significant overall. The explanation comes from the choice, all processes start from the best individual every 500 generations.

5.2 Evaluation with Existing Standards

This section provides an evaluation that we built to compare the optimized grids generated by our system and the popular existing grids such as PODD. First, we describe the evaluation set up and what will be involved. Then we show and analyze the obtained results.

5.2.1 Evaluation Description

Many applications or physical supports are generally using the well known pictogram grids such as PODD or Proloquo2Go grids[24], considered as the standards and very efficient. We want to situate our optimized grids compared to them. The main purpose is to evaluate two features : the cost of the grid $Cost(G)$ (introduced in section 3.3) and the number of mismatches within the standards. A mismatch corresponds to a missing pictogram in the grid during the cost computation (mismatch rate for the distance cost). During the $Cost(G)$ computation, we add 0 to the cost for a mismatch. To obtain $Dist(G)$ and the mismatches, we select an evaluation text corpus. We also set the Movement Cost C_M and the Selection Cost C_S to 1. To evaluate the coherence, instead of considering $Cohe(G)$, we compute the average word similarity $Cohe(G)_{Avg}$ because words might be different between grids. So we want to know the coherence of each grids independently from the other (page size, number of pictograms, etc.).

To compute the cost of the standard grids, we need to convert the grid in our file format. However, some grids are organized in a graph structure such as the PODD, but the cost computation is done by using the page tree structure and tree-based algorithms. Therefore, we convert the grid structure into a tree by cutting some links. This is a choice of complexity we make and it may impact the pictogram grid efficiency for the evaluation. Furthermore, we remove Meta-Pictograms and Sentence pictograms because we could not manage them yet. Nevertheless, we keep the Polysemic pictograms by assigning to each synonym, one representative word.

5.2.2 Results

We present the results of the evaluation we conduct between the grids we optimized and a PODD. The PODD 15 we selected is a french version containing pages of dimension 5x3 and 1 398 pictograms. First, we perform an exhaustive evaluation with our two set of grids (AT and Tcof). Then, we evaluate the PODD and our Tcof$_{Hybrid}$ with more or less complex corpora.

5.2.2.1 Approximate Evaluation with a French PODD 15

We compare the grids we generated by using our optimization pipeline, with a French PODD 15.

Small grids : AnimalTexts

The first evaluation aims to observe how small grids generated from a simple text can be efficient. Therefore, we establish the comparison between the PODD and the optimized grids from the AnimalTexts we built. The corpus is the same as the generation step, because it is not impactful here. Table 5.2 presents the results of the evaluation for the AT grids.

		Costs					
Grids	Picto/Pages	Dist(G)	%Δ_{Dist}	Cohe(G)$_{Avg}$	Cost(G)	%Δ_{Cost}
AT	AT$_{Random}$	104 / 5	6.31	-48.14	-0.32	1.301	-24.31
	AT$_{Dist}$	111 / 5	**5.89**	**-66.24**	-0.31	1.210	-38.62
	AT$_{Hybrid}$	104 / 5	5.92	-64.96	-0.36	**1.208**	**-38.90**
	AT$_{Cohe}$	104 / 5	6.19	-54.43	**-0.44**	1.246	-33.31
	French_PODD	1 398 / 114	6.97	∅	-0.42	1.422	∅

Table 5.2: Evaluation results between the AnimalTexts dataset associated grids and the french PODD 15 with$\alpha = 0.5$. **Dist(G)** represents the distance cost, %Δ_{Dist} the distance difference between the PODD and aa grid, **Cohe(G)**$_{Avg}$ the average coherence of a pictogram within the grid, **Cost(G)** the grid cost and %Δ_{Cost} the cost difference between the PODD and a grid.

First, our small AT corpus regroups rather elemental and common sentences about animals. However, the French PODD has a mismatch rate of 74.35% and 75 words from AnimalTexts are missing. Determinants are not included in the PODD, but common words such as *"pingouin"* or *"fromage"* do not belong in it. However, the mismatch rate is representative because it may also depend on the conversion quality of the PODD (some words exists in an other form or may includes undesirable characters). Then, we notice all of AT grids have a lowest $Dist(G)$ and $Cost(G)$ than the PODD, even the random one. Due to size of the grid, as there are too few pictograms and pages, the path between each pictograms will be usually shorter. The $Cohe(G)_{Avg}$ is lower than the PODD and does not depend on the size and the structure of the grid. Because the corpus we built deals only with animals, the whole text has already a thematic. Nevertheless, AT$_{Cohe}$ is significantly lower than the three others and reflects the influence of the coherence cost with $\alpha = 1$. Furthermore, the grids having respectively the best $Dist(G)$ and the best $Cost(G)$ are AT$_{Dist}$ and AT$_{Hybrid}$ as expected.

Large Scale : TCOF - Philosophy

Next, we evaluate the generated grids from the TCOF Philosophy part (section 4.1.1) and the french PODD. Here, the evaluation corpus is the TCOF Longitudinal part from the Adult-Child section (41 965 words for 3 594 lines) of the TCOF.

		Costs					
Grids	Picto/Pages	Dist(G)	%Δ_{Dist}	Cohe(G)$_{Avg}$	Cost(G)	%Δ_{Cost}
TCOF	Tcof$_{Random}$	2 482 / 100	12.34	-2.22	-0.28	2.620	+6.17
	Tcof$_{Dist}$	2 500 / 100	11.80	-43.43	-0.27	2.504	-18.72
	Tcof$_{Hybrid}$	2 500 / 100	**11.80**	**-43.73**	-0.33	**2.489**	**-21.48**
	Tcof$_{Cohe}$	2 483 / 100	12.47	+10.19	-0.39	2.625	+7.39
	French_PODD	1 398 / 114	12.37	∅	**-0.42**	2.594	∅

Table 5.3: Evaluation results for the TCOF dataset and associate grids with the TCOF - Longitudinal part with $\alpha = 0.5$. **Dist(G)** represents the distance cost, %Δ_{Dist} the distance difference between the PODD and aa grid, **Cohe(G)**$_{Avg}$ the average coherence of a pictogram within the grid, **Cost(G)** the grid cost and %Δ_{Cost} the cost difference between the PODD and a grid.

The French PODD has a mismatch rate of 55.89% and 1 532 missing words where our grids have 11.40% of mismatches and 761 missing words. The high mismatch rate from the PODD tells us that there are less words commonly and frequently employed in french sentences, than in our grids.

Table 5.3 presents evaluation results between our Tcof grids and the french PODD. First, excepted Tcof_{Cohe}, all of our grids have a lower $Dist(G)$ than the PODD, even the random one. The main reason comes from the transformation of the structure we performed on the PODD. Originally, the PODD is organized in a graph structure, but we transformed it into a simplified tree for choices of complexity and to be able to apply the algorithms we previously described (section 3.3.2). As a graph has no notion of depth, by transforming it, the maximal depth of the converted PODD in tree is 4 while our grids do not exceed 2. More levels increase the total distance because of the successions of up-downs and the euclidean distance of directory pictograms to select. In other words, pictogram-grids using a tree structure are simplified versions but are faster to generate. Here, after transformation, the PODD might loose optimizations provided by its graph structure.

In this simplified situation, the grid with the lowest $Dist(G)$ and $Cost(G)$ is the Tcof_{Hybrid}. However, the PODD is still better than our Tcof_{Cohe} regarding the $Cohe(G)_{Avg}$. The Tcof_{Cohe} is rather close to it and as we explained in the experiments results, the fitness did not converge yet. So, by performing more iterations of the process, we may be closer to the PODD or even exceed it.

For now, we only evaluated our grids with one French PODD 15. Later, we should perform an evaluation with other standards such as different PODD versions or Proloquo2Go grids.

5.2.2.2 Diverse Dataset

Next, we perform the evaluation with different dataset to have a better comparison with other text corpora. We extracted and preprocessed ESLO-Jeunes, a corpus from the ESLO dataset [10] constituted of interviews with young people representing 1 266 lines and 27 052 words. We also randomly selected 69 texts from the TCOF Adult-Adult part (section 4.1.1) to constitute another corpus. Table 5.4 regroups evaluation results for each corpus :

	TCOF-Transversal		**TCOF-Adult (69 texts)**		**ESLO-Jeunes**	
Grids	Tcof_{Hybrid}	PODD	Tcof_{Hybrid}	PODD	Tcof_{Hybrid}	PODD
Dist(G)	12.69	13.29	11.43	12.11	10.70	11.39
Cost(G)	2.681	2.793	2.409	2.537	2.251	2.382
% Mismatch	**9.96%**	54.40%	21.51%	74.20%	22.12%	74.05%
Missing Words	2 126	3 329	3 592	4 871	1 687	2 607

Table 5.4: Evaluation results with different text corpus. TCOF-Transversal, 69 texts from the TCOF-Adult section and the ESLO-Jeunes are the three implied corpora. **Dist(G)** represents the distance cost, **Cost(G)** the grid cost, **% Mismatch** the mismatch rate.

First, for every datasets, our grid has the lowest cost $Cost(G)$. The explanation is the same as before, the structure of the PODD is modified and might impact its efficiency. Nevertheless, the mismatch rate increases for both grids as the texts become more complex. The TCOF-Adult and ESLO-Jeunes corpora constitute a more advanced vocabulary than the TCOF-Philosophy or TCOF-Transevrsal. Our grid has about 22% of mismatches for those datasets, but only

9.96% of mismatches for the TCOF-Transversal. We also observe the difference for the PODD with 54.40% for the child corpus and about 74% for transcriptions from older people discussions.

Eventually, we could evaluate our grids with more texts or other types of datasets such as stories, book extracts or articles.

5.3 Discussion of Results

This section summarizes the results we obtained and the acquired knowledge.

5.3.1 Small-Scale and Large-Scale Optimization

We have generated several grids by varying the α coefficient but also the text corpus. We noted that results depend mainly on the quality and the size of the dataset.

With small-scale corpora such as AnimalTexts, we obtained grids satisfying the targeted displays and observed a fast convergence of the fitness. The execution time for the optimization of small grids is also shorter than the optimization of larger grids.

However, with large-scale datasets like the TCOF Philosophy part, the optimized grids, including about 2 500 pictograms, do not completely fulfill the different display criteria.

First, we observed that sizeable distance grids such as Tcof_{Dist} are mostly influenced by the words occurrences as the computation of $Dist(G)$ requires euclidean distances and shortest paths in a tree. Frequent words will tend to be at the *"center"* of the grid, thus in the top-left of the pages. In the end, the grid is optimized in terms of distance between pictograms, but is less similar to an expected semantic-syntactic display. Furthermore, the final display depends on the dataset quality and preprocessing. Later, we should considerate different approaches such as bi-grams or predictive models to enhance the syntactic impact and obtain displays that are close to our expectations.

Then, even after many generations, Tcof_{Cohe} and Tcof_{Hybrid} fitness did not converge. Thematic partially emerged from pages and the global coherence is not enough. The coherence is computed by vector similarities and mostly influenced by the language model (fastText). We obtained pages regrouping only verbs due to the Part of Speech (POS), only numbers or only animals close to Thematic displays, but also incoherent pages. In particular with hybrid grids, we observed the language model we used limits the expected contextual display behavior (pages without adjectives or verbs, etc.). To improve the coherence part, we might try to vary α and test different language models.

Finally, small-scale optimizations are rather promising while improvements (language models, grid evaluation cost, dataset preprocessing and quality, parameters, etc.) will be required to better handle larger grids and be closer to the desired displays.

5.3.2 Pictogram Grid Evaluation with Standards

The evaluation we performed between our grids and the French PODD 15, revealed the efficiency potential of the small grids and situated how close were the optimized grids with regards the standards. It also showed that the complexity of the language used to generate and optimize our grids, may limit its usage with other texts.

First, the cost of small grids is obviously lower than the PODD because there are far less pages and pictograms, so the total distance to build a pictogram sequence is shorter. But it shows how fast and we can use smaller grids and how efficient they are when optimized.

Then, the evaluation between the PODD and our larger optimized grids showed acceptable results. However, this was an approximation as we transformed the PODD structure into a tree. Some PODD optimizations may have disappeared because of this operation and we can not ensure that our grids are preferable.

The mismatches informed us about the grid vocabulary status (size, language complexity, etc.) and the capacity to build a pictogram sequence or not, limited by the dataset choice and the preprocessing step.

In the end, to get a better insight of the grid efficiency, we need to modify our tree-based approach into a graph-based approach to fit with the standards and not to have an approximate comparison. We also learned that the quality of the corpus and how we transform it impact the grid vocabulary and the capacity to build pictogram sequences.

— 6 —
Conclusion and Perspectives

6.1 Conclusion

In our study, we explore the automatic generation of optimized pictogram grids. The main challenge is to submit a solution that generates pictogram grids to satisfy as much as possible the user's needs and to be used in an efficient way. We come up with a pipeline, based on preliminary works, starting from a raw text corpus, generate random grids and optimize them by using a genetic algorithm with multi-processing. We preprocess corpora by using a language model to obtain filtered sentences corresponding to the language of the pictograms. As we want to solve a minimization problem, the genetic algorithm we implemented follows a fitness representing the cost of the grid. We introduce a new approach to compute the cost of a grid based on two separated part - the distance for the syntax impact and the coherence for the semantic impact - with a coefficient to vary the type of display we want. The more important is the distance part, the more the display will be influenced by the syntax and vice-versa. We also present adapted genetic operators to pictogram grids, regrouping four different mutations and one crossover. To observe, analyze and navigate through the output grids from the pipeline, we introduced a new vizualiation tool called PictoGriz.

To test our approaches, we generate grids from different corpora. Small-scale grids are faster to optimize and are organized in an expected display whereas larger grids show promising results but revealed some limits of our current approach. Then, we conduct an evaluation to situate our grids with standards such as the PODD. Nevertheless, results are approximate as we apply transformation on the PODD structure. Finally, the evaluation of the grids abilities to form pictogram sequences, with corpora having a more or less advanced language, show that a pictogram grid is also impacted by the quality of the sentences and their complexity.

In the end, the automatic optimized pictogram grid generation is relatively recent and complex due to the plurality of required parameters and constraints. Our work followed and enhanced the first contributions about this subject and led to optimistic results. Nevertheless, this project should be pursued as there are still improvements to do. Next section proposes different perspectives and approaches that should be explored in the optimized pictogram grid generation field.

6.2 Perspectives

A way to make use of our approaches in real application is to focus on small-scale optimizations. Experiments showed satisfying results and realistic execution times with a small corpus. Our proposition is to select a word and extract sentences from a dataset that contain this word. Then, by applying our pipeline on the extracted corpus, we generate and optimize its corresponding pictogram grid. In the end, we could obtain in a few minutes, an optimized pictogram grid related to the word.

For now, our grids are designed in a page-tree structure (see section 3.2) mainly for complexity reasons. However, from the evaluation we performed, we realized that standard grids like the PODD or Proloquo2Go corresponds to a graph (because of the links between directory pictograms) and we only approximate the comparison. Later, we should convert our approach to be suitable with graphs (distance computation with a Dijkstra algorithm, graph-based mutations, etc.).

From our experiments, we also noticed that the execution time of our algorithm is rather long. This is mainly due to the sequential process we established. There are many operations we can parallelized with the use of GPUs. For instance, when we evaluate the population, we wait for the cost computation of the x_i individual before starting the cost computation of the x_{i+1} individual. This step and the genetic operators should be done in parallel to reduce the execution time.

We might also improve the genetic algorithm. First, by making it adaptive to avoid to fall in local minima and be more efficient. We already suggest adaptive mutations in Appendix A.2. Then, we can define other adequate genetic operators such as different crossovers or new mutations.

Another perspective we considered is to get rid of the Genetic Algorithms because of its different disadvantages (execution time, local minimum, etc.) and apply a predictive model such as PictoBERT (or a derived version of BERT if we work with words). An idea of approach is to build the neighborhood of a selected pictogram, by predicting the next K candidates. For instance, if the pictogram is the word *Mouse*, the predictive model will output a pictogram list such as *[Cheese (62%), Computer (37.5%), Tree (0.5%)]*, for $K = 3$, and store those pictograms closer to the pictogram in the grid, as the probability is higher.

Finally, up until now, we only worked on transcriptions and texts. For future works, we should automatically generate pictogram grids directly from speech to correspond with reality. The task will involve audio processing and we may consider language models such as Wav2Vec [23] for tokenization. Resources built upon multilingual texts could be handled as well by using language models like mBart [14] to align languages and generate pictogram grids including a vocabulary from any target language.

— A —
Appendix

A.1 One page structure and cutting

At the beginning of our researches, we explored a simplification of the representation of the grid before the optimization process. The idea is to have only one page including all pictograms. Next, we applied the whole pipeline on it. The resulting page is then optimized, but too large as it contains the entire vocabulary. We post-processed the grid by cutting the final pages. We thought about different approaches to automatically cut pages in a smart way.

A.1.1 Similarity-based cut

As the grid should be optimized, the similarity-based approach is to cut N * M pages based on neighborhood similarities of selected pictograms. However, there are some issues. First, we may miss optimization information that belongs to the neighborhood of the selected pictograms (N*M). So we might cancel contributions from the genetic algorithm step. Then, we used a language model at the end of the optimization process to cut pages, which made our optimization pipeline less impactful in the whole process.

A.1.2 Proximity Score cut

The proximity score cut is a clustering approach. We calculate the distances between pictograms and clusters within the optimized grid and considerate the bi-gram occurence probability in the corpus as a weight to compute a proximity score. We can compute the proximity score Pc_{ik} with the following formula :

$$Pc_{ik} = dist(p_k, p_i) * \frac{(p(w_k, w_i) + p(w_i, w_k))}{2} \tag{A.1}$$

where $dist(p_k, p_i)$ is the distance between two pictograms p_k and p_i in the page, $p(w_k, w_i)$ and $p(w_i, w_k)$ are respectively the occurrence probabilities of the bigrams (w_k, w_i) or (w_i, w_k) in the text.

The main issue is the starting point. We do not have any idea of how many clusters (pages) we need and which pictogram should be a center.

Cutting is a major problem regarding the organization of pages and the graph/tree structure. We won't be sure to have an optimized grid by adding links and append another dimension.

Thus we might need another optimization step after the cut that take into account the structure information.

A.2 Detailed Adaptive Mutations Process

Considering K mutations, in the initial step (i=0), we associate to each a probability P_k the corresponding mutation triggered, such as $\sum_{k=0}^{K} P_k = 1$. Our goal is to update the probability depending on the fitness minimization contribution of the the mutation during a step. To do so, we compute the mutation contribution rate η, by making the difference between the step i and the step $i+1$:

$$\eta = \frac{f(x_{i+1}) - f(x_i)}{f(x_i)} \tag{A.2}$$

with f(x), the fitness of an individual and $\eta \in [0,1]$.
Then, thanks to η, we update the mutation probabilities.

If the mutation did not happen, the update will reduce its probability :

$$P_k = P_k - (P_k * \eta) + P_k * \sigma \tag{A.3}$$

with η, our mutation contribution rate, σ a parameter to avoid a vanishing probability (set to 0.01 for instance). We want to penalize the mutation probability, but not loose the mutation forever and make sure to never obtain $P_k = 0$ or $P_k = 1$.

If the mutation happened, we increase its probability depending on its contribution :

$$P_k = P_k + ((1 - P_k) * \eta) - \sum_{i=0}^{K} P_i * \sigma \tag{A.4}$$

where $P_i \neq P_k$.

A.3 AnimalTexts Sample

We provide few sentences from the AnimalTexts (AT) corpus we built. AT regroups basic sentences describing animal characteristics.

Chat et Souris :

- *Le petit chat veut manger la souris.*
- *Elle aimerait engloutir un bout de fromage sans avoir peur.*
- *On est content car il observe les oiseaux.*

Animaux :

- *Un dauphin est un mammifère marin.*
- *Le poisson-chat vit en eau douce.*
- *La girafe vit dans la savane.*

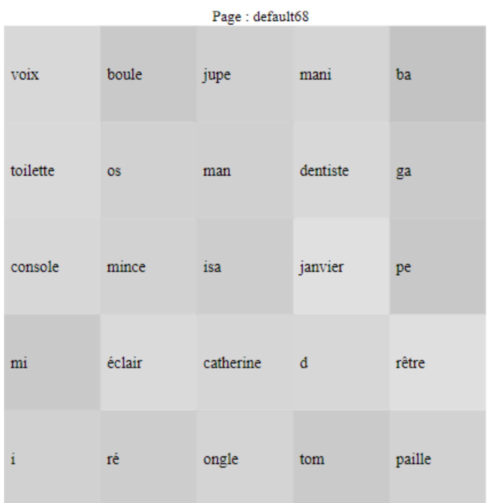
A.4 Execution Time for the Grids Optimization

Table A.1 shows the different execution times for each optimization. Because of the different processes, we averaged the time for one generation to finish and we also averaged the total time.

Grids	s/gen	Nb. Generations	Total Time
AT_{Dist}	0.21	3 000	≈13min
AT_{Cohe}	0.13	3 000	≈8min
AT_{Hybrid}	0.19	3 000	≈12min
$Tcof_{Dist}$	27.29	16 000	≈128h
$Tcof_{Cohe}$	3.68	16 000	≈20h
$Tcof_{Hybrid}$	27.56	16 000	≈133h

Table A.1: Execution times for each optimization. The CPU used for the experiments is a 16 core Intel Xeon E5-2623 v4 @ 2.60GHz. The AnimalTexts corpus regroups 312 words and a vocabulary size of 99 for the AnimalTexts, where the TCOF-Phhilosophy part regroups 60 151 words and a vocabulary size of 2 402. Optimization times of larger grids are significants as regards the smaller grids from the AnimalTexts corpus.

A.5 Page Examples from the Optimized Grids



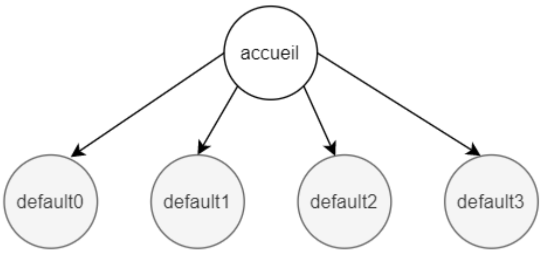
(a) Similarity heatmap of the page "default68" from $Tcof_{Hybrid}$ grid. This page includes a low coherence. Some words are just syllables or rarely used nicknames.

(b) Similarity heatmap of the "default39" page from $Tcof_{Hybrid}$ grid. This page includes a high coherence. Almost every words of the page are verbs. This is an example of part of speech clustering from the language model.

Figure A.1: Contrasted similarity heat maps from the $Tcof_{Hybrid}$ grid : a) Page "default68" with a low coherence. b) Page "default39" with a high coherence.

A.6 AT$_{Hybrid}$ Grid in Details

We provide the entire AT$_{Hybrid}$ we optimized.

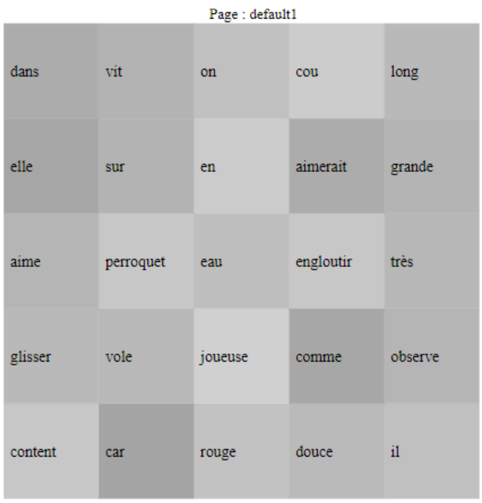


(a) Page-tree of the AT$_{Hybrid}$ grid.

(b) Heatmap of the page "accueil" from AT$_{Hybrid}$.

Figure A.2: Page Tree and page "accueil" of the AT$_{Hybrid}$ grid.

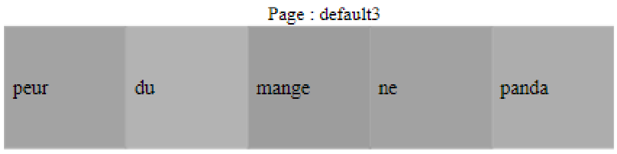
(a) Heatmap of the page "default0" from AT$_{Hybrid}$. We observe in top-left "la", and following feminine words "souris,mer,lionne,baleine,loutre,glace" close to it.



(b) Heatmap of the page "default1" from AT$_{Hybrid}$. We observe mainly determinants and verbs in this page due to the POS. With "dans" in the top-left.

Figure A.3: Pages "default0" and "default1" from AT$_{Hybrid}$.

(a) Heatmap of the page "default2" from AT_{Hybrid}. We observe that "les" is close to "feuilles, bananes, reptiles, oiseaux", which are all of the plural words in the vocabulary.



(b) Heatmap of the page "default3" from AT_{Hybrid}. This page contains only 5 words.

Figure A.4: Pages "default2" and "default3" from AT_{Hybrid}.

Bibliography

[1] Michael Bender and Martin Farach-Colton. The lca problem revisited. volume 1776, pages 88–94, 04 2000.

[2] D.R. Beukelman and J.C. Light. *Augmentative & Alternative Communication: Supporting Children and Adults with Complex Communication Needs*. Paul H. Brookes Publishing Company, Incorporated, 2020.

[3] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606*, 2016.

[4] Mike Bostock. D3.js - data-driven documents, 2012.

[5] Lucie Chasseur, Marion Dohen, Benjamin Lecouteux, Sébastien Riou, Amélie Rochet-Capellan, and Didier Schwab. Evaluation of the acceptability and usability of augmentative and alternative communication (AAC) tools: the example of pictogram grid communication systems with voice output. In *ACM SIGACCESS 2020 - Conference on Computers and Accessibility*, ASSETS '20. The 22nd International ACM SIGACCESS Conference on Computers and Accessibility - Virtual Event, page Article 68, Athènes, Greece, October 2020.

[6] K. R. Chowdhary. *Natural Language Processing*, pages 603–649. Springer India, New Delhi, 2020.

[7] Kenneth De Jong. *Evolutionary Computation – A Unified Approach*. 01 2006.

[8] Ulrich Derigs, Martin Kabath, and Markus Zils. *Adaptive Genetic Algorithms: A Methodology for Dynamic Autoconfiguration of Genetic Search Algorithms*, pages 231–248. Springer US, Boston, MA, 1999.

[9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2018.

[10] Iris Eshkol-Taravella, Olivier Baude, Denis Maurel, Linda Hriba, Céline Dugua, and Isabelle Tellier. Un grand corpus oral « disponible » : le corpus d'orléans 1 1968-2012. 01 2011.

[11] Virginie André et Emmanuelle Canut. « mise à disposition de corpus oraux interactifs : le projet tcof (traitement de corpus oraux en français) ». pages 147–148, 2010.

[12] X. Han and Zhao Zhang. Topological optimization of phononic crystal thin plate by a genetic algorithm. *Scientific Reports*, 9:8331, 06 2019.

[13] Annu Lambora, Kunal Gupta, and Kriti Chopra. Genetic algorithm- a literature review. In *2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon)*, pages 380–384, 2019.

[14] Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. Multilingual denoising pre-training for neural machine translation. *CoRR*, abs/2001.08210, 2020.

[15] M.J. Mahmoodabadi and A.R. Nemati. A novel adaptive genetic algorithm for global optimization of mathematical test functions and real-world problems. *Engineering Science and Technology, an International Journal*, 19(4):2002–2021, 2016.

[16] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space, 2013.

[17] George A. Miller. *WordNet: A Lexical Database for English.* Communications of the ACM Vol. 38, No. 11: 39-41. 1995.

[18] E.C. NEGRE. *Communiquer autrement: Accompagner les personnes avec des troubles de la parole ou du langage*. APPRENDRE ET RÉAPPRENDRE. De Boeck Supérieur, 2017.

[19] Barry S. Oken, Umut Orhan, Brian Roark, Deniz Erdogmus, Andrew Fowler, Aimee Mooney, Betts Peters, Meghan Miller, and Melanie B. Fried-Oken. Brain–computer interface with language model–electroencephalography fusion for locked-in syndrome. *Neurorehabilitation and Neural Repair*, 28(4):387–394, 2014. PMID: 24370570.

[20] Jayr Pereira, David Macêdo, Cleber Zanchettin, Adriano Oliveira, and Robson Fidalgo. Pictobert: Transformers for next pictogram prediction. *Expert Systems with Applications*, 202:117231, 04 2022.

[21] G. PORTER. *Pragmatic Organization Dynamic Display communication books : Direct access templates*. Melbourne, Australia: Cerebral Palsy Education Centre (CPEC), 2007.

[22] Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. Stanza: A Python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, 2020.

[23] Steffen Schneider, Alexei Baevski, Ronan Collobert, and Michael Auli. wav2vec: Unsupervised pre-training for speech recognition, 2019.

[24] Samuel Sennott and Adam Bowker. Autism, aac, and proloquo2go. *Perspectives on Augmentative and Alternative Communication*, 18:137–145, 12 2009.

[25] Gilles Sérasset. DBnary: Wiktionary as a Lemon-Based Multilingual Lexical Resource in RDF. *Semantic Web – Interoperability, Usability, Applicability*, 6(4):355–361, 2015.

[26] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to sequence learning with neural networks, 2014.

[27] VINCENT VANDEGHINSTE, INEKE SCHUURMAN LEEN SEVENS, and FRANK VAN EYNDE. Translating text into pictographs. *Natural Language Engineering*, 23(2):217–244, 2017.

[28] Carlos Vargas and Didier Schwab. Automatic generation of communication grids for people with motor and intellectual disabilities using artificial intelligence methods. June 2021.

[29] Céline Vaschalde, Pauline Trial, Emmanuelle Esperança-Rodier, Didier Schwab, and Benjamin Lecouteux. Automatic pictogram generation from speech to help the implementation of a mediated communication. In *Conference on Barrier-free Communication*, Geneva, Switzerland, November 2018.

[30] Tonio Wandmacher, Jean-Yves Antoine, Jean-Paul Departe, and Franck Poirier. SIBYLLE, an assistive communication system adapting to the context and its user. *ACM Transactions on Accessible Computing* , 1(1):1–30, 2008.

Other Sources :

[31] PROPICTO (PRojection du langage Oral vers des unités PICTOgraphiques) project :
https://www.propicto.unige.ch/project-description/

[32] AugCom, AAC application of the InterAACtionBox project :
http://www.augcom.net/

[33] TCOF : Traitement de Corpus Oraux en Français, ATILF, ORTOLANG (Open Resources and TOols for LANGuage), 2018 :
www.ortolang.fr - https://hdl.handle.net/11403/tcof/v1

[34] Comité International Olympique, Tokyo 2020 dévoile les pictogrammes des Jeux :
https://olympics.com/fr/infos/tokyo-2020-devoile-les-pictogrammes-des-jeux , 2021-03-30.

[35] American Speech-Language-Hearing Association (n.d.), Augmentative and Alternative Communication (Practice Portal). :
www.asha.org/Practice-Portal/Professional-Issues/Augmentative-and-Alternative-Communication/.