FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

# Intelligent Form Design: Exploiting Unstructured Natural Language Fields

**Diogo Filipe da Silva Yaguas**


Master in Informatics and Computing Engineering

Supervisor: Henrique Lopes Cardoso, PhD

Second Supervisor: Luís Paulo Reis, PhD

July 21, 2021

# Intelligent Form Design: Exploiting Unstructured Natural Language Fields

**Diogo Filipe da Silva Yaguas**

Master in Informatics and Computing Engineering

July 21, 2021

# Abstract

Every year, public and private institutions receive numerous contacts through online forms. Typically, these forms have structured and unstructured textual fields, which means that the information filled by the user is often incomplete and inconsistent, preventing the proper processing of the contact within the institution. Usually, the user is unaware of the internal processing that the contact has downstream, whether human or semi-automatic and the underlying data needs. For these reasons, the contact intent of the user with the institution may not be appropriately addressed.

The concept of usability in forms is very much directed to the user experience in the front-end. User experience is a central aspect of Human-Computer Interaction processes, being decisive for the interaction between the user and the institution. Intelligent Forms aim to create a new conception of usability to develop methods that improve communication effectiveness between the user and the institution.

Natural Language Processing techniques allow the application of tools that help the user fill in unstructured fields with less noise, supporting the correct expression of the purpose of the contact with the target institution.

Moreover, institutions have large amounts of data with information about previous communications and the result of their processing. This data can be used in machine learning models capable of capturing the processing that is done in the specific domain of that institution. This exploration of domain-specific machine learning classification models may be used to ensure that the information necessary for processing the contact is, in fact, provided.

The Electronic Complaints Book and the contact form of the Portuguese Economic and Food Safety Authority served as case studies to explore new features, using text classification models, that upgrade form usability and permissibility, with the aim of improving the quality of submitted contacts. With the analysis of these case studies, it was possible to define minimum requirements for the development of Intelligent Forms. Through this, we have generalized and built a tool that allows designing Intelligent Forms dynamically.

The tool was used to build intelligent versions of the forms for the case studies mentioned, which were later evaluated and tested to understand if it was possible to improve communication between users and institutions and increase the chances that the contacts made can be subsequently processed. Our findings indicate that there is effectively a reduction in the limitation in communication between users and institutions. This is due to the assistance provided to users both in filling fields semi-automatically, as well as in classifying the content in order to pre-process the contact. Nevertheless, it is necessary to continue working to improve the concept of Intelligent Forms. It will be essential to implement new features, as well as the use of better performing machine learning models that allow obtaining high classification results.

**Keywords**: Assistive Writing Tools, Classification Models, Human-Computer Interaction, Intelligent Forms, Intelligent Interfaces, Machine Learning, Natural Language Processing

# Resumo

Todos os anos, instituições públicas e privadas recebem inúmeros contactos através de formulários online. Normalmente, esses formulários possuem campos textuais estruturados e não estruturados, o que significa que as informações preenchidas pelo utilizador muitas vezes são incompletas e inconsistentes, impedindo o correto processamento do contacto dentro da instituição. Geralmente, o utilizador não tem conhecimento do processamento interno que o contacto possui a jusante, seja humano ou semiautomático, e das necessidades de dados subjacentes. Por esses motivos, a intenção de contacto do utilizador com a instituição pode não ser abordada de forma adequada.

O conceito de usabilidade em formulários é muito direcionado à experiência do utilizador no front-end. A experiência do utilizador é um aspeto central dos processos de Interação Humano-Computador, sendo decisiva para a interação entre o utilizador e a instituição. Os Formulários Inteligentes visam criar uma nova conceção de usabilidade para desenvolver métodos que melhorem a eficácia da comunicação entre o utilizador e a instituição.

As técnicas de Processamento de Linguagem Natural permitem a aplicação de ferramentas que auxiliam o utilizador no preenchimento de campos não estruturados com menos ruído, auxiliando na expressão correta do propósito do contacto com a instituição de destino.

Além disso, as instituições possuem grande quantidade de dados com informações sobre comunicações anteriores e o resultado do seu processamento. Esses dados podem ser utilizados em modelos de aprendizagem computacional capazes de capturar o processamento que é feito no domínio específico daquela instituição. Esta exploração de modelos de classificação de aprendizagem computacional de domínio específico pode ser usada para garantir que as informações necessárias para processar o contato sejam, de facto, fornecidas.

O Livro Eletrónico de Reclamações e o formulário de contacto da Autoridade de Segurança Alimentar e Económica serviram de caso de estudos para explorar novas funcionalidades, recorrendo a modelos de classificação textual, que atualizam a usabilidade e permissibilidade, com o objetivo de melhorar a qualidade dos contactos apresentados. Com a análise desses casos de estudo, foi possível definir requisitos mínimos para o desenvolvimento de Formulários Inteligentes. Com isso, generalizamos e construímos uma ferramenta que permite desenhar Formulários Inteligentes de forma dinâmica.

A ferramenta foi utilizada para construir versões inteligentes dos formulários dos casos de estudo citados, que foram posteriormente avaliados e testados para entender se era possível melhorar a comunicação entre utilizadores e instituições e aumentar as chances de os contactos realizados serem posteriormente processados. Os resultados indicam que há efetivamente uma redução da limitação na comunicação entre utilizadores e instituições. Isto deve-se ao auxílio prestado aos utilizadores tanto no preenchimento de campos de forma semiautomática, bem como, na classificação do conteúdo de forma a pré-processar o contacto. No entanto, é necessário continuar a trabalhar no sentido de aprimorar o conceito de Formulários Inteligentes. Será imprescindível a

implementação de novos recursos, bem como a utilização de modelos de aprendizagem computacional de melhor desempenho que permitem a obtenção de resultados de classificação altos.

**Keywords**: Aprendizagem Computacional, Ferramentas de Auxílio à Escrita, Formulários Inteligentes, Interação Humano-Computador, Interfaces Inteligentes, Modelos de Classificação, Processamento de Linguagem Natural

# Acknowledgements

First and foremost, I want to dedicate this work to my father, who, despite not being with me physically, will always be in my heart. I know I would have unconditional support in all the decisions I made that brought me to where I am today.

I want to start by thanking Professor Henrique Lopes Cardoso for all the patience, effort, and availability he had at all times. Thank you for allowing me to take part in this research and carry it out.

To the women of my life. Starting with my mom, the strongest woman I know, who always made an effort to not miss out on anything, a thank you that I know is not enough to thank for everything she did for me. To my godmother, who always made sure I had everything I needed, even if far from me. To my grandmother, who helped make me the man I am today. Without them, I wouldn't be where I am.

To my brothers. Daniel and David, thank you for always supporting and believing in me, even when they didn't know they were doing it.

To my hometown friends. To Beatriz, Bernardo, Casalta, Fontes, Joana, Marcela, Mariana Melo, Mesquita, and Ricardo for believing in me and always showing in me what I couldn't see. Without them, I hadn't followed the path I took.

To the friends I made in these 5 years. Starting with those who accompanied me on this journey, Carolina, Daniel, Gonçalo, Inês, and Teresa. Without you guys dealing with my dramas and cheering my days when I needed them most, these years hadn't been the same. Thank you for all the time we spent together. Next, I would like to thank the friends whose friendships were created at the most random times and who have always supported me ever since. Thank you, Cristiana, Inês, and Joana, for accompanying me on this journey.

To AEFEUP and all the friendships created. I especially want to thank those who made me grow as a person. Brandão, Xana, and Xavi thank you for all the fun we had together and for the moments spent in and out of that building.

Finally, to Rui Pedro. Thank you for always being there for me when I needed it most. Thank you for believing in me, even when I couldn't. Thanks for showing me what I was capable of. Without you, I wouldn't have done a work of this level, and it would undoubtedly have been more difficult.

Diogo Filipe da Silva Yaguas

*"If you believe in your dreams, I will prove to you,*
*that you can achieve your dreams just by working hard."*

Rock Lee, Naruto

# Contents

# List of Figures

# List of Tables

# Abbreviations

| | |
|---|---|
| AI | Artificial Intelligence |
| ASAE | Portuguese Economic and Food Safety Authority |
| AUC | Area Under the Curve |
| BERT | Bidirectional Encoder Representations from Transformers |
| CF | Contact Form |
| CNN | Convolutional Neural Networks |
| ECB | Electronic Complaints Book |
| GPT | Generative Pre-trained Transformer |
| HCI | Human-Computer Interaction |
| IF | Intelligent Form |
| LM | Language Model |
| ML | Machine Learning |
| NER | Named-entity recognition |
| NLP | Natural Language Processing |
| POS | Part-of-Speech |
| RNN | Recurrent Neural Networks |
| ROC | Receiver Operating Characteristic |
| SLM | Statistical Language Models |
| SVM | Support Vector Machine |
| TF-IDF | Term Frequency-Inverse Document Frequency |

# Chapter 1

# Introduction



A contact form (CF) allows data and information to be filled, which permits the formalization of communications with organizations, such as companies or state institutions. Nowadays, it is increasingly common to switch from a physical format to a digital one, thus evolving from traditional to online forms. These are software developed, allowing their processing in organizations through computer networks, reducing paper use. This type of form is prevalent among organizations that use internet resources, making CF available on their websites. Many important tasks require people to fill forms, such as filing tax returns, applying for business permits, reporting financial compliance, requesting health care data [1], and contacting private and public institutions in order to make complaints, requests for information, or compliments. These contacts are then processed manually or semi-automatic to follow them up, which requires several inputs related to the contacted institution's activity domain. The processing is often hampered since many contacts end up not containing the information necessary for their proper processing within the institution concerned. This happens due to the poor filling of structured and unstructured fields, usually without automatic support mechanisms, and often leads to the need for later interactions with the user in order to obtain additional information. When this is not possible or not convenient, the contact is simply filed without further processing.

Natural Language Processing (NLP) is developing rapidly due to the growing interest in human-computer communication, plus the availability of big data, powerful computing, and enhanced algorithms. NLP can support computers in communicating with humans in their language and expand other language-related tasks. For example, NLP allows computers to read a text, interpret it, and determine which parts are essential. Moreover, techniques are being applied to handle written information produced in contacts between users and institutions. With the increasing use

of Artificial Intelligence (AI) technologies, devices will be able to understand users better, offering enhanced user experience and responding to user needs.

Many of the main technological innovations in the field of Human-Computer Interaction (HCI) that have appeared in recent years are evolutions of resources that already existed that intelligent interfaces have replaced. The purpose of these is to integrate intelligent automated skills into HCI where the main impact is a HCI that increases performance or usability in critical ways, boosting skills and capabilities so that human performance with an application excels [2].

This dissertation aims to create an Intelligent Form (IF), which is an intelligent CF, capable of improving communication quality between a user and an institution, providing the user with different tools that support it in the correct filling of the CF.

## 1.1 Problem Statement and Motivation

When people contact an institution to request information, make a complaint or denunciation, or even acknowledge a well-provided service, they face several problems that can hinder the contact's purpose and lead to poor CF usage. This is due to the fact that the user is unaware of the internal processing that the contact has downstream, whether human or semi-automatic, and the underlying data needs. In addition, these problems lead users to give up on formalizing a contact, making communication between the user and the institution even more difficult.

A key component of Human-Computer Interaction procedures is user experience, which is crucial to the user-institution relationship. The idea of usability in forms is heavily focused on the front-end user experience. Intelligent Forms aims to develop ways that increase communication effectiveness between the user and the institution by developing a new conception of usability.

An institution that is the target of this study and serves as a motivation for carrying out this work is the Portuguese Economic and Food Safety Authority (ASAE). This institution is the administrative authority in Portugal specialized in the areas of food security and economic surveillance [3], whose goal is to prevent and monitor compliance with the existing laws that oversee legislation related to economic activities in the food and non-food sectors and report possible risks related to the food chain.

Denunciations and complaints made directly to ASAE are submitted through the CF available on their website or through e-mail. In the last five years, the number of complaints made annually through these means was around twenty thousand. This high number of complaints is stored in a database for later processing. The problem with this processing is that many complaints do not have enough information or refer to problems that are not part of the ASAE's jurisdiction. This makes it essential to improve the communication between the user and the institution [4].

The Electronic Complaints Book is intended for consumers and users who wish to submit complaints in a dematerialized way, exercising their right to complain electronically. The digital version became mandatory and transversal to all sectors of the national economy at the beginning of 2020, leading to a significant increase of digital complaints. For these reasons, the electronic

version of the book also serves as a case study and a motivation for carrying out this work, as it is quite permissive, and many users face difficulties when filling a complaint.

## 1.2 Objectives and Research Questions

The objectives, acting as milestones toward the principal goal of the study, are as follows:

- Exploration and formulation of a set of requirements as the foundation for developing the concept of Intelligent Forms, based on the analysis of the case studies.
- Development of planned functionalities for the IF.
  - Detection of inter-field inconsistencies.
  - Autocomplete of structured fields.
  - Machine learning for domain-specific free-form field classification.
- Development of a tool that allows the construction of an IF.
- Comprehensive functional assessment, at least partially based on ASAE's and Electronic Complaints Book's case studies.

The following questions were formulated, taking into account the purpose of our study:

1. How can NLP tools be used to validate inter-dependencies between structured and unstructured fields and facilitate the (semi-)automatic filling of structured fields in an IF?
2. How can we take advantage of domain knowledge to improve communication effectiveness between the user of a CF and the respective organization/institution?

## 1.3 Document Structure

This dissertation is structured in 6 additional chapters:

- Chapter 2: covers the main concepts for this study, focusing on the development of forms from the perspective of Software Engineering and Intelligent Interfaces, writing assistance tools, and domain-specific text classification models that enable checking inter-field inconsistencies.
- Chapter 3: describes the general requirements that an IF should fulfill, identifying user roles and the kinds of functionality available for each one.
- Chapter 4: explanation of the solution created for the concept of IF, presenting the developed IF tool and the use cases.

- Chapter 5: presents the experiments carried out for the case studies, the datasets used, and the details of the experiments. As well features the results obtained from the conducted experiments and contains the analysis of those results.
- Chapter 6: presents the main conclusion of this work. It analyzes the approaches used to achieve the objectives and suggests potential lines of work that might be employed to develop further or to include more features.

# Chapter 2

# Literature Review



This chapter introduces the background and definition of basic concepts in this project's context. We start by presenting, in Section 2.1, Intelligent Interfaces and the development of forms from the perspective of Software Engineering. Next, Section 2.2 introduces tools for assistive writing, namely, predictive typing, autocomplete and spelling/grammar checking from the perspective of NLP. Finally, in Section 2.3, we discuss the use of domain-specific machine-learning classification models to validate if an unstructured field has enough information to allow processing of the contact.

## 2.1 Intelligent Interfaces

Intelligence is a human characteristic that has been defined and redefined over time, with no concrete and general definition. However, with the advancement of technology, humans increasingly try to impose human characteristics on machines, and intelligence is one of the most tried.

User interfaces are studied in a computer science field called Human-Computer Interaction. These days, with the advance of technology, tasks such as speech and language processing, conversational agents, and dialogue systems [5] allow the appearance of a new type of interactive and straightforward user interfaces to simplify the interaction between computers and the user. However, several users reported problems and negative experiences they encountered in interactions with various intelligent systems, thus becoming a challenge recognized by the HCI field [6]. Therefore, there is a need to create Intelligent Interfaces that can anticipate and adapt to users'

needs, learn new concepts and techniques, take the initiative, and make suggestions [7]. With the help of AI and HCI, such types of interfaces are being developed.

Intelligent Interfaces are human-machine interfaces that intend to improve the performance, effectiveness, and naturalness of HCI [8]. Better performant interfaces mean that the user will accomplish the desired task in less time. Effectiveness in an interface allows the user to achieve its intent, customizing the content and form of the interaction to the user's context. Finally, naturalness allows the interplay between the user and the interface to be as human-like as possible through speech or writing. Figure 2.1 show the different fields associated with II.

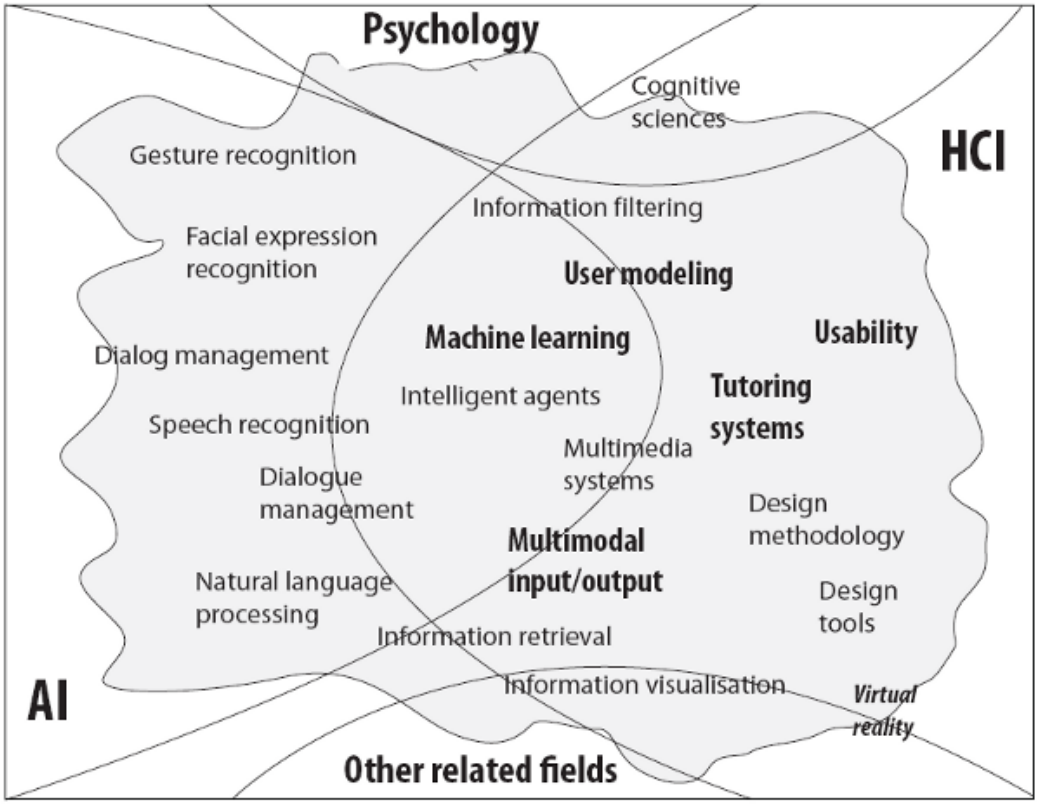


Figure 2.1: Intelligent Interfaces and associated fields (from Alvarez-Cortes et al. [7])

### 2.1.1 The key components

The concept of intelligence in an interface depends on the user who uses it. The components necessary for an interface to be considered intelligent are described by Sarang Shaikh et al. [9]:

1. *System Functionality*: the interface must interact with the system and have information about the tasks that the user intends to perform.

2. *Intelligent about the user*: the interface should facilitate communication with the user, applying input and output methods that are natural to the user.

3. *Interface must be sensitive to user needs*: the interface must understand when the user needs help with certain tasks generating hints and tips.

Different components are considered in the development of Intelligent Interfaces considered efficient. User behavior analysis is an essential functionality because it enables the interface to understand or at least conjecture the user's intention and release the user from tedious tasks often required when using a non-adaptive interface [10]. User modeling is one of the fundamental components of Intelligent Interfaces construction. This component supports the interface to help the user, both asking the right questions at the right time, making suggestions, and directing the user in the right way, taking into account the user's final goal.

Another component that can be considered necessary is plan-based interfaces. These are driven by the notion that by evaluating a user's actions in relation to a user model that contains potential intentions and goals, the interface may deliver a more intelligent and cooperative class of system responses. [11]. This process gathers user intentions based on the user's tasks in a software application and uses the history of users' actions to inform the user of suitable and adequate decisions. For many applications requiring interaction between humans and computer systems, it would be highly desirable for machines to converse in English or other natural languages familiar to their human users [12]. Natural Language is one of the best methods for creating compelling and automated interfaces.

Almost any natural communication among humans requires multiple, concurrent modes of communication. Undoubtedly, any HCI system that aspires to have the same naturalness should be multimodal [13]. In the interface, the use of various communication methods is called multimodal communication. For example, a mouse, natural language written text by typing, or voice input can be some of these methods. Because the communication method is determined based on its input and output, whether it be text, vision, or sound, this component is always dependent on the user model.

### 2.1.2 The different types of Intelligent Interfaces

Intelligent Interfaces try to solve problems that standard interfaces cannot solve or handle suitably. Consequently, there are numerous types of attempts to develop Intelligent Interfaces with different performances and objectives.

Alvarez-Cortes et al. [7] states that Direct Manipulation User Interfaces replace low-level general-purpose user interface styles like command languages with more concrete, meaningful, transparent, and valuable techniques concerning the human user's intentions, knowledge, and skills. In this type of interface, the user models are used to predict the user's goals or tasks or identify the users inputs patterns. An example of this type of interface is UNIX commands' prediction by Davison and Hirsh [14]. Intelligence can be adjusted in different approaches taking into account the predictions. For example, the system can predict and speculate the user's action and initiate a pre-execution. Then, if the user frequently performs the same set of actions, the

system can execute that set as a whole or make these actions available quickly, offering quick assistance [15].

Moving away from these more traditional Intelligent Interfaces and using a Machine Learning (ML) approach, it is possible to identify two types of interfaces that act as intermediaries between the user and the direct manipulation interface. On the one hand, we have Informative Interfaces that can visualize essential information, for example, task progress, user performance, task difficulty, email-filtering, or recommender systems. From a form perspective, these interfaces can also help users complete the form, considering that they can monitor the content filled in, directing the user to fill it out correctly.

This information can be beneficial as it can improve human decision-making [16]. This interface attempts to select or modify information and display only those elements that the user may be interested in or that are beneficial for the task at hand. The user's feedback involves marking and classifying the recommended options as desirable or undesirable in these systems. However, since the user must provide feedback, this type of interface is obtrusive and misleads the user from the central task.

On the other hand, we have Generative Interfaces that allow us to see an interface design unique to everyone, based on user data. This system's primary purpose is to improve data quality by reducing data entry time for a human. In addition, these interfaces support improved feedback because the user can disregard a recommendation and substitute it with another one. The feedback is linked to the interface's interaction. In the context of CF, these interfaces can use methods that allow the automatic filling of fields, whether structured or unstructured, thus reducing the amount of information that the user has to enter.

#### 2.1.2.1 Decision-Making

Intelligent user interfaces are distinguished by their ability to respond to a complex adaptation approach at run-time and make multiple communication choices on "what", "when", "why" and "how" to interact. The characterization of decision-making is defined by the adaptation determinants, constituents, goals, and rules. Stephanidis et al. [17] affirms that the adaptation determinants used in decision-making usually involve user and information characteristics and the task performed.

The HCI field has been extensively debating and studying the use of models as a fundamental part to expand our human perception of the processes related to user interface adaptation. Moreover, there is a relevant distinction between a model, architecture, and the goals sought by them.

Existing systems use specific rules based on a predetermined determinant group to adapt to various predefined components to achieve predetermined goals. In short, the adaptation process is not flexible and cannot be easily transferred between applications.

#### 2.1.2.2 Mixed-Initiative Approach

A Mixed-Initiative system engages an innovative method, where the system and the user interact to produce a personalized interface. Expressly, users are provided with an interface mechanism that gives them full control over the customization process and adaptive support to help them customize their interfaces effectively [18]. This type of interface enables users and intelligent agents to collaborate efficiently [19], requiring flexibility that allows the participants to interact naturally to make their contributions.

From a user perspective of a contact form, using this kind of approach can improve how the user interacts with the interface. The form in collaboration with a human user can help formalize a contact, providing guidance and suggestions on the contact by evaluating user intentionality, event importance, and causality to detect filling problems.

### 2.1.3 Intelligent Forms Interfaces

The concept of Intelligent Interfaces is very much directed towards the concept of adaptive interfaces, that is, adjusting to different users in order to make the experience more pleasant for each one. However, what is sought in an Intelligent Interface in the sense of a CF is a reactive interface, that is, one that is able to react to the content filled in by the user in the various fields, analyzing it and checking for the possibility of inconsistencies, lack of information or possible errors.

It is possible to show or hide fields or parts through Conditional Logic, a feature pervasive in contact form design, for instance, to allow or disable the submit button of a form dependent on a previous user selection or input. However, as it does not accept variables and data other than field values, Conditional Logic can only support basic conditions containing only field values [20].

The idea of IF is the easy creation, maintenance, and evaluation of user-friendly CF that can give real-time feedback [1]. Furthermore, during form filling, the IF may provide the users with efficient and context-sensitive cognitive assistance in the form of auto-completion lists, disabled invalid options, guidance and alerts about breaches of restrictions, and automatic filling fields with appropriate values to meet complex constraints.

Accordingly, an IF meets Adaptive Interfaces, as it requires the interface to grasp the user's purpose and free the user from repetitive tasks by helping and directing the user in the right way, taking into account the user's final goal. An example of AI technology that can be applied is a rules engine that enables information resources to be outsourced and provides the user with a series of evidence-based, context-specific suggestions that are created and can be applied at run-time [21].

## 2.2 NLP in Intelligent Forms

For many people, computer capacities to deal with language as handily as humans praise the introduction of really insightful machines. This conviction's premise is that language's successful utilization is entwined with our overall intellectual capacities [5].

A human coordinates a computer system as a servant in traditional HCI, telling it what to do and how to do it. The user is the one who contains all the objectives in traditional interfaces and is the one who leads in the discovery of ways to satisfy those objectives. Therefore, a prime priority of efficient software is to enrich the user experience by simplifying the writing process.

### 2.2.1 Natural Language Processing and the different tasks

NLP is a subfield of computer science, AI, and linguistics dedicated to developing human language technology's capacity. It is a theoretically motivated range of computational techniques for one or more linguistic analysis levels to analyze and represent naturally occurring texts in order to achieve human-like language processing for different tasks [22]. NLP allows the technology to understand the user. This complex system can extract information from the contact with humans.

AI is the simulation of human intelligence in machines programmed to think like humans, mimic their actions, and allow complex dialogues between machines and humans. NLP is indispensable to allow the machine to understand what is being said or written and structure the best response.

With the rising use of AI technologies, it is increasingly vital that devices understand users, offering better experiences and responses to their needs. Therefore, NLP researchers intend to gather information on how people interpret and use language and establish effective methods and strategies for computer systems to understand and exploit natural language to accomplish the desired tasks [23].

Applications that involve NLP cover a vast range of tasks. A list of the main applications that use NLP is as follows:

- *Named-entity recognition (NER)*: recognize entities in the text, such as persons, locations, organizations and dates.

- *Sentiment Analysis*: in contexts such as consumer surveys and social media ratings where people share their thoughts and suggestions, sentiment analysis is most helpful. The result can be a numerical score in more complicated situations and separated into as many categories as needed.

- *Text Summarization*: it is used in situations such as news stories and articles on science. Text summarization methods can be grouped into extractive and abstractive methods. The extractive text summarization technique entails extracting key points from the source document and combining them to create a summary. The abstraction method involves paraphrasing and compressing sections of the source document.

- *Aspect Mining*: involves identifying aspects or features of an entity and forming opinions about those aspects. It is a text classification method that evolved from Sentiment Analysis and NER.

- *Terminology extraction*: the most used and most essential words and phrases are extracted automatically from a text. It helps summarize the substance of texts and acknowledge the critical subjects covered.

#### 2.2.1.1 Text preprocessing

Preprocessing is an essential step for text classification applications. Text data is not completely clean in most instances. Data from multiple sources have varying characteristics, making text preprocessing one of the most significant phases in the classification process. Preprocessing the text involves placing the text into a form that can be evaluated for a task. This preprocessing can be accomplished through several techniques:

- *Tokenization*: the method of splitting string sequences into individual words, keywords, phrases and symbols. It also interprets and groups isolated tokens to create higher-level tokens [24].

- *Noise removal*: delete characters, numbers and parts of text from characters that can mess with the text analysis. The reduction of noise is one of the most important preprocessing measures for text, being strongly domain-dependent. Some noise removal must occur before other preprocessing actions.

- *Lowercasing*: while often disregarded, text data's lowercasing is one of the easiest and most effective ways of text preprocessing. Despite its beneficial property of decreasing sparsity and vocabulary size, lowercasing may have a detrimental effect by increasing complexity and ambiguity [25]. For example, when filling out a contact form, it is sometimes necessary to indicate the location. In the Portuguese language, the words "Porto" and "porto" have different meanings. Lowercasing both make them similar, allowing the classifier to lose major predictive.

- *Stopwords removal*: stopwords are commonly used words. Removing stopwords can be applied in applications such as search engines and document classification, where keywords are more important than generic phrases, but there are applications, such as searching for precise quotes, where stopwords can be substantial. A list of terms can be arranged by frequency to find the stop words and choose the high frequencies according to their lack of semantic meaning [24].

- *Lemmatizing/Stemming*: lemmatization replaces a word's suffix with another one or entirely eliminates a word's suffix to get the lemma. On the other hand, word stemming usually does not generate a simple form but rather an approximation of this form called stem [26]. However, this can come at the expense of neglecting considerable syntactic nuance [25].

Following the syntactic representation of the word, the text will be translated into a vector containing the frequency of terms as the next step in text preprocessing. Even though intuitive,

this approach s constrained by the fact that specific words commonly used in the language may dominate such representations [27].

The bag-of-words model is among the most widely used representation methods for text classification. This technique is effortless and versatile and can be used to extract features from texts in various ways. Nonetheless, it suffers from high-dimensional sparsity and a lack of associations between words [28]. A text is represented as the multiset of its terms in this model, avoiding grammar and even word order while preserving multiplicity.

On the other hand, Term Frequency-Inverse Document Frequency (TF-IDF) may be used to weigh down frequent corpus words. TF, where each word is mapped to a number equivalent to the number of instances of that word in a text, is the most straightforward weighted word function extraction method. IDF is a measure of whether a term is common or rare in a given document [27].

### 2.2.2 Language Models

Language Models (LM) is an AI model that has been trained to predict the next word or words in a text-based on the preceding words.

The models are prepared by studying a language's characteristics and features to predict words. The algorithms then interpret text data by passing it through an algorithm that establishes context rules in natural language. Following that, the model uses the generated rules in language tasks to predict or generate new sentences.

A variety of probabilistic methods are used for training a LM. Such methods differ depending on the reason for which a LM is built. For example, the sum of text data to be evaluated and the math used for research creates a difference in the approach taken to a LM to create and train.

There are primarily two types of LM. In order to learn the frequency of word distribution, Statistical Language Models (SLM) use standard statistical methods such as N-grams, Hidden Markov Models and some linguistic rules. Alternatively, Neural Language Models are in the NLP field and have exceeded the SLM productivity. To model language, they use various forms of neural networks.

#### 2.2.2.1 Statistical Language Models

Statistical models include the development of probabilistic models that can predict the next word in the sequence given the words that precede them. There are already several SLM in use. N-Gram is one of the easiest language modeling methods. A probability distribution is generated for a sequence of $n$, where it may be any number and the size of the sequence of words assigned to a probability is specified. In probability computation and compact representation types such as finite-state automaton, the standard n-gram LM allows fast factorization, making it ideal for large-scale distributed serving [29].

#### 2.2.2.2 Neural Language Models

Neural Language Models are based on neural networks and are frequently viewed as an advanced approach to NLP tasks. Google's BERT (Bidirectional Encoder Representations from Transformers) is one of the most popular LM. BERT executes a deep bidirectional representation by observing both left and right contexts. With only one extra output layer and minor changes, the model can be fine-tuned [30].

Another powerful LM is the Generative Pre-trained Transformer (GPT) model. These LM can perform a task with few or no examples of the task's type and perform equivalent or even better than state-of-the-art models trained in a supervised method. Radford et al. [31] suggested using unlabeled data to learn a generative LM and then calibrating the model by giving examples of particular downstream tasks. GPT-1 showed that the LM acted as an efficient pre-training target that could further generalize the model. The design allowed transfer learning and, with a little fine-tuning, could execute multiple NLP activities. This model demonstrated generative pre-training strength and allowed the introduction of more comprehensive datasets and more parameters for other models that might release this capacity better.

The GPT-2 model innovations primarily concerned using a broader sample to learn an even better LM and adding more parameters to the model. Radford et al. [32] claimed that the log-linear fashion output improved with the model's power increase. However, the reduction in the perplexity of LM still demonstrated no saturation and continued to decrease with the number of parameters.

With 175 billion parameters, the GPT-3 model was built and required no calibration and just a few demonstrations to understand and execute tasks. There were 100 times more parameters than GPT-2 on this model. It performs well on downstream NLP tasks due to the vast number of parameters and comprehensive dataset GPT-3 has been trained on, [33].

These models are indeed robust LM and have revolutionized the world of NLP by using only the instructions and few samples to execute a multitude of tasks.

### 2.2.3 Assistive Writing Tools

Assistive writing tools employ ML to help users through various writing process steps, including auto-completion, predictive typing, and spelling/grammar checking. These tools support NLP to analyze text and provide suggestions or related content. These tools can handle the otherwise complex and slow writing content process, allowing users to write more quickly and confidently.

Several tools allow assisting the user when it comes to writing, such as:

- *Predictive typing*: aimed to give proposals that decrease writers' typing effort by offering shortcuts to enter a small number of words that the system predicts are destined to be typed straightaway [34].

- *Auto-completion*: provide real-time, interactive suggestions to help users compose texts quickly and with confidence. Furthermore, cuts back on repetitive idiomatic writing via providing immediate context-dependent suggestions [29].
- *Spelling/grammar checking*: efforts to assist users in improving their text by using a combination of rules and statistics for checking errors in spelling, punctuation, word choice, and sentence structure [35].

#### 2.2.3.1 Predictive typing

Predictive language modeling in text input interfaces was first developed to assist those with motor impairments and poor typists. These predictions are generated from a previously-entered text model created and maintained adaptively [36].

Word prediction helps people enter text more rapidly and precisely [37]. In systems where the user presents a complete pronouncement to it, improvements in speed and accuracy have been reported, and the system can then process it as a whole, such as speech recognition systems that allow users to fix errors using various techniques [38] or systems that enable a sentence-at-a-time approach [39].

Arnold et al. [34] studied how predictive text systems affect how we type. The researchers ran experiments asking participants to write descriptive captions for photographs, finding that text became more concise, linear, and less vivid when individuals employ these structures. This research provides convincing evidence that intelligent systems developed to increase human work productivity often influence human work content and do so in unexpected ways.

Many other systems rely on the prediction of words and phrases, hence increasing their ability to help [40], in the same way, systems that suggest something that was previously written and recently eliminated by the user [41]. However, some of these systems have limitations that can lead to problems, such as high-quality suggestions in phrases suggestions systems since these are hard to generate or tend to suggest words or phrases with a positive character, leading to bias when these systems are used in reviews or complaints.

#### 2.2.3.2 Auto-completion

Auto-completion systems are applied in several areas: search engines, email writing, command line suggestions, code completions in IDE's, assisting people with writing disabilities, speeding up the text simplification method, and giving full control over information preservation to users. In addition, these systems help those who write since they consider what the user is writing and suggest the rest of the sentence as the user types it, therefore becoming a system quite similar to the previous one.

Many methods of NLP need LM to develop tools of this kind. Perhaps one of the most popular and widely used is Gmail's Smart Compose, a system that gives real-time, interactive suggestions to aid users in compose messages rapidly [29]. Figure 2.2 shows an example of a suggestion made by Gmail's Smart Compose method.

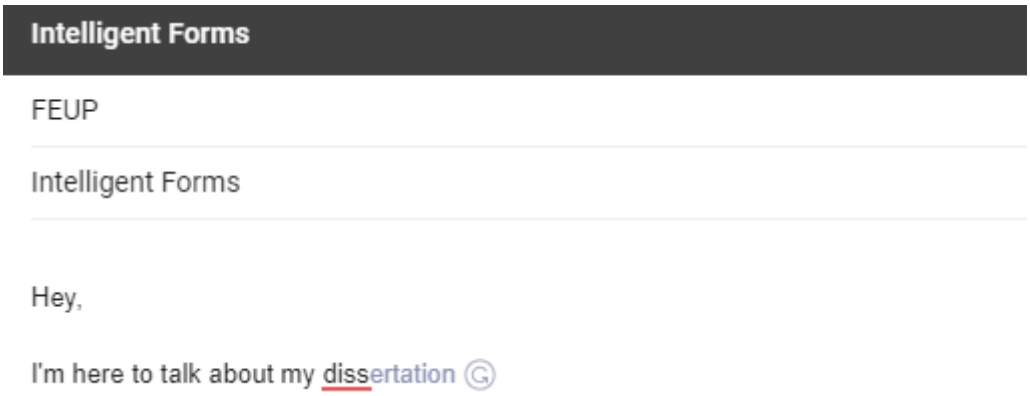


Figure 2.2: Smart Compose screenshot

In addition to this system, Gmail provides a quick response service, named Smart Reply, that generates semantically diverse suggestions that can be used as complete email responses [42].

Some studies concluded that auto-complete systems aid users while searching, improving the search result [43]. Furthermore, this tool can mislead the user while typing, so this kind of suggestion needs to be relevant to compensate for the distraction that may cause to the user [44]. Additional research found that the suggestions should appear during the user inputs and not only in the end, considering they can lead to the more ubiquitous use of suggested terms [45].

#### 2.2.3.3 Spelling/Grammar Checker

Spell and grammar checkers are currently an essential component of several computer software such as text processors and web browsers. These systems help to write a text by checking each word for its correctness, assisting users in improving their text using a union of rules and statistics for checking errors in spelling, punctuation, word choice, and sentence structure [35].

First, there is a need to define the types of errors. A non-word error happens when a word in a source text is represented as a string that in a given word list or dictionary does not fit any actual word. When a source-text word is represented as a string that appears in the dictionary but is not identical to the source-text word, a real-word error occurs [46].

Over the years, comprehensive research has been conducted on spell and grammar checking technology. Most spell-checking methods described use dictionaries as a list of correct spellings. These spellings can help the algorithm find the target word.

Techniques based on dictionaries are still used [47], despite having a performance limitation due to their architecture. In order to measure the difference between the desired word and the possible correct terms of the dictionary, these approaches apply comparison reference metrics. For this reason, the more extensive the dictionary, the longer the calculation time and the lower the performance.

Another method employed is affix-stripping. The main idea is to store a stem word and apply a set of rules to deal with affixes and change the stems if vital. The strategy was compelling

in recognizing incorrect spellings, yet it neglected to recommend reasonable words, as, in this method, nonexistent words were frequently suggested.

Real-word error is hard to recognize using these systems because the misspelled word exists in the word reference. However, by taking context into account, these misspellings become recognizable. The use of confusion sets, a small group of words that are likely to be confused with one another, is a solution to this problem, treating the problem as a disambiguation task [48].

As technology advances, users have easy and free access to online grammar checker programs, such as SpellCheckPlus, Grammarly and Ginger Software. These systems are designed to detect linguistics errors in user's texts and provide automated and prompt feedback.

Grammar checkers receive inputs in different ways. However, most of these systems only check sentences, defined through punctuation. Before scanning for errors, they separate the text corpus into a set of sentences, then each sentence in the sentence list is preprocessed using various methods to verify the veracity. Bhirud et al. [49] defines these methods, representing them in an algorithmic form:

- *Sentence Tokenization*: this includes tokenization of words and segmentation of sentences. The sentence is tokenized into words, followed by breaking them down into constituent morphemes and populating the word from the lexicon with lexical details.
- *Morphological Analysis*: word stems and related affixes are returned through morphological examination
- *Part-of-Speech (POS) tagging*: categorizing each word in a text in correspondence with a particular POS tag, depending on the definition of the word and its context.
- *Parsing Stage*: tests the syntactic constraints between input word and the construction of the input sentence's hierarchical phrasal/dependence structure using the selected approach/methodology. In case of any problem, the grammatical error flag also provides the user with the auto-correction method or existing recommendation list.

In general, three methods of grammar checking are used. As seen in Naber [50], the Rule-based approach is a traditional solution to grammar testing since it manually constructs grammar rules. Specialists in linguistics design these high-quality rules. The approach seems straightforward since a rule is easy to add, edit or remove. Despite this, writing rules require an exhaustive knowledge of the underlying language's grammar. Rule-based programs can provide a comprehensive overview of flagged errors, rendering the system particularly useful for computer-aided language learning purposes. Nevertheless, it is challenging to maintain hundreds of grammar rules manually.

The statistical approach uses an annotated corpus to generate a POS tag list. Some sequences will be prevalent from these generated sequences, and others will probably not occur at all. In other texts, commonly occurring sequences will also be deemed correct and unusual sequences will lead to errors. The availability of a significant amount of POS-annotated corpus is one of the

major prerequisites for applying this approach. Additionally, the POS tags used must reflect the grammatical properties required to check the underlying language's agreement [51].

ML is the most common grammar testing tool at present. Since the solution to ML is based on probabilistic performance, it leaves considerably fewer formal guarantees [52]. These approaches use an annotated corpus that is in turn used to automatically identify and correct grammar errors and perform predictive analysis on the language. Compared to rule-based methods, the failures produced by these systems are difficult to describe. As they are entirely dependent on the underlying corpus, ML-based systems do not require extensive grammar knowledge. However, the non-availability of a large annotated corpus prevents using such techniques for grammar control. The results also depend significantly on how clean the corpus is [53].

A combination of ML and rule-based techniques creates a hybrid method that can improve the system's performance, addressing a wide range of problematic errors.

## 2.3 Domain-specific Text Classification Models

Classifying contacts into distinct categories can have a significant impact on contact processing. Using ML to categorize contacts enables evaluators to handle processing quicker and more efficiently by selecting a set of contacts that matches their domain.

In order to validate whether a field contains the necessary and sufficient information, it is necessary to resort to classification models that can assign tags or categories to text according to its content, being one of the main tasks in NLP with extensive applications. In addition, a classification model attempts to draw some conclusions from observed values.

Text can be a rich source of knowledge, but it can be difficult and time-consuming to obtain insights from its unstructured nature. ML text classification can help automatically organize and interpret text, efficiently and cost-effectively, to simplify processes and boost data-driven decisions.

One of the functions most commonly carried out by so-called Intelligent Systems is supervised classification. A training dataset is fed into the supervised ML classification algorithm.

### 2.3.1 Machine Learning algorithms

There are various text and document classification algorithms available. Next, some ML algorithms used to classify text are discussed.

#### 2.3.1.1 Logistic Regression

One of the most pioneering methods of classification is Logistic Regression. This method defines and describes the link between one or more categorical or continuous predictor variables and a categorical outcome variable [54]. The logistic regression goal is to train from the probability of output being 0 or 1 given an input. In order to predict categorical outcomes, the Logistic Regression

classifier works well. Notwithstanding, this prediction involves each data point's independence, which attempts to predict results based on a set of independent variables [27].

#### 2.3.1.2 k-Nearest Neighbors

A non-parametric technique used for classification is the k-Nearest Neighbors algorithm. This classifier's basic concept is to find among the training documents the $k$ closest neighbors of the candidate document and score the categories of the $k$ closest neighbors.

In the case of an extensive training set, the algorithm compares the test document with every training sample, decreasing performance. Additionally, the acceptable choice of k-value and a suitable similarity function also affects the algorithm's efficiency.

#### 2.3.1.3 Decision Tree

A decision tree is a structure comparable to a tree that classifies the data by tracing the root-to-leaf direction. The attribute with the most significant information gain is selected as the parent node and the child node is allocated to the subsequent attributes. Each branch expresses the result of the test and the class label is represented on the leaf node where the classification is carried out [55]. The key downside of this classifier is that even a slight variance in the information will lead to a different class prediction and can easily overfit the model [56]. This problem can be solved using decision tree pruning.

#### 2.3.1.4 Naive Bayes Classifiers

The Naive Bayes classifier, a generative model, has been commonly used for its versatility in both the training and classification processes [57] and are among the most used algorithms in text classification and text analysis. The most basic version of Naive Bayes Classifiers was developed using TF, a feature extraction technique that counts the number of words in a document. Bayesian classifiers assign the most likely class to an example described by its feature vector [58].

While it is less precise than other discriminatory approaches, numerous researchers have shown that it is efficient enough at text categorization in several domains [59]. Naive Bayes is based on Bayes' Theorem, which computes the conditional probability of the occurrence of two events based on the probability of each case's event. Furthermore, these models allow each attribute to participate equally and separately from other attributes towards the final decision, where computational utility is excellent relative to other text classifiers.

The need for relatively limited amounts of training data is advantageous for Naive Bayes. As long as the right category is more probable than the other categories, the Bayesian classification provides satisfactory results. Thus, there is no need to accurately estimate the odds allocated to categories [60].

#### 2.3.1.5 Support Vector Machine

Support Vector Machine (SVM) is the most commonly used pattern classification technique based on ML techniques available today. Initially, this method was designed for binary classification tasks. In order to provide precise results, SVM such as Naive Bayes does not need much training data. However, this algorithm takes more computing resources than Naive Bayes, but the outcomes are faster and more precise. SVM distinctively afford balanced predictive efficiency, even in studies where sample sizes can be small, due to their relative simplicity and versatility to address various classification problems [61].

SVM maps the input vector to a higher dimensional space where a maximum hyperplane separation is created. This classifier's main objective is to find a maximum hyperplane separation between vectors belonging to a group and vectors not belonging to it [62].

SVM falls into the intersection of the kernel and classifiers of wide margins. Feature selection, time series analysis, reconstruction of a chaotic system, and non-linear key components have been applied to SVM [63].

#### 2.3.1.6 Deep Learning algorithms

Deep Learning is a group of algorithms and techniques inspired by neural networks. These algorithms attempt to leverage the unknown structure in the input distribution to identify good representations, frequently at multiple levels, with higher-level learned features specified in terms of lower-level features.

They focus on building large neural networks, and many methods deal with extensive datasets of labeled analog data, such as text. The most popular deep learning algorithms are: *Recurrent Neural Networks*, *Convolutional Neural Network*, *Long Short-Term Memory Networks*, among others.

##### 2.3.1.6.1 Deep Neural Networks

Deep Neural Network architectures are programmed to learn through multiple layer connections where every single layer only receives a link from the previous one and provides connections in the hidden part only to the next layer. The input layers are built using various feature extraction methods such as TF-IDF and word embeddings, and the output layer represents the number of classes [64].

**Recurrent Neural Networks:** Another neural network model tackled by researchers for text classification is Recurrent Neural Networks (RNN). This method specifies weights to the previous series data points. Hence, it is a valuable tool for the classification of text. In RNN, the neural net considers previous nodes' data that permits for a better semantic interpretation of the structures in the dataset [65]. Although RNN is helpful for text classification problems, it may suffer from vanishing gradient problems.

**Convolutional Neural Networks:** Another Deep Learning architecture used for hierarchical text classification is Convolutional Neural Networks (CNN). While initially designed to process images, CNN was often used effectively to classify text [66]. In order to extract different characteristics, CNN includes convolutional operations of moving frames or windows that evaluate and minimize different overlapping regions in a matrix. The ability to bootstrap words embedded in this form of a neural network also makes it an ideal candidate for information extraction and non-annotated text classification [67].

### 2.3.2 Evaluation of models

It is just as vital to evaluate a ML model as it is to develop one. These models are developed to perform with newly, previously unknown data. As a result, creating a robust model necessitates a complete and varied evaluation.

Before evaluating the model, there is the need to understand how many predictions the model got right and the model failed. A confusion matrix is not a measure for evaluating a model, but it does give information about the predictions, as seen in Table 2.1.

| | | **Predicted values** | |
|---|---|---|---|
| | | Positive (P) | Negative (N) |
| **Actual values** | Positive (P) | True Positive (TP) | False Negative (FN) |
| | Negative (N) | False Positive (FP) | True Negative (TN) |

Table 2.1: Confusion matrix for binary classification.

A true positive result is one in which the model accurately forecasts the positive class. On the other hand, a true negative is an event in which the model adequately predicts the negative class.

A false positive is an outcome in which the model forecasts the positive class erroneously. Conversely, a false negative is an outcome in which the model forecasts the negative class incorrectly.

With these results the model evaluation can be performed:

- *Accuracy*: measure the percentage of texts that have been labeled with the right class.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{2.1}$$

- *Precision*: measures the percentage of accurate predictions by considering only the cases identified as belonging to the interest class by the model.

$$Precision = \frac{TP}{TP+FP} \tag{2.2}$$

- *Recall*: measure the percentage of positive examples known as such.

$$Recall = \frac{TP}{TP+FN} \tag{2.3}$$

- *F1 Score*: the harmonic mean of precision and recall. Because it considers both false positives and false negatives, the F1 score is a more helpful metric than accuracy for situations with unequal class distribution.

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{2.4}$$

A Receiver Operating Characteristic (ROC) curve is a graph that depicts a classification model's performance by combining confusion matrices at all threshold values. The X-axis of the ROC curve is the True Positive Rate, which measures the proportion of positive class that is correctly predicted as positive, and the Y-axis of the ROC curve is the False Positive Rate, which measures the proportion of negative class that is correctly predicted as negative.

Instead of finding the optimal threshold value on the ROC curve, another metric called Area Under the Curve (AUC) can be used. This metric aggregates performance across all classification criteria. AUC is the area under the ROC curve between (0,0) and (1,1), calculated using integral calculus. For example, AUC may be interpreted as the likelihood that the model rates a random positive case higher than a random negative example.

## 2.4 Summary

The purpose of this chapter was to present the background of the fields connected with this research as well as the work established in previous phases. It started with a research about what an Intelligent Interfaces is, what constitutes it and the different types that exist. With this analysis, it was possible to understand what will be necessary and what is dispensable to include in an Intelligent Form.

Then, with the research for tools to assist writing, namely, predictive typing, autocomplete and spell/grammar checkers, it was possible to understand the best ways to introduce this type of mechanism in a form.

Lastly, research was carried out to recognize the possibility of introducing ML classification models, understanding the different types of models that exist and their advantages and disadvantages. Their integration in forms allows the validation of information in unstructured fields.

# Chapter 3

# Intelligent Forms Requirements Specification



This chapter includes four sections and describes the general requirements that an Intelligent Form (IF) should fulfill. Section 3.1 demonstrates the purpose and scope of this concept. Section 3.2 offers an overview of the concept's functionality. This section also identifies the user roles and the kinds of functionality available for each one. Section 3.3 provides the specification of requirements in precise terms and describes the different interfaces. Section 3.4 presents the prioritization of requirements.

## 3.1 Introduction

An IF is a CF that empowers the user with tools that increase communication efficiency with the institution. A description of the scope and an overview is provided in this section.

An IF is a web-based application that will upgrade the typical CF. It provides advanced and intelligent features that allow the form to validate unstructured field information, endowing the user with tools that enable intelligent assistance in filling out the form. The form's design and construction will consist of selecting the desired fields and complementing them with models that will allow for intelligent assistance of the kinds detailed in this requirements specification.

The application can have several forms of use. For example, it can be an independent website or a form embedded in an institutional website, both of which allow its use from a web browser

or an app, either from a computer or a mobile device. Furthermore, the user should be able to intuitively fill the form and understand whether the information provided in the different structured and unstructured fields is congruent and consistent. Additionally, the user will have access to several tools that assist in writing the various unstructured fields and that, optionally, automatically fill in the structured fields.

## 3.2 Overall description

An overview of the IF concept will be given in this section. First, the different users and the features associated with each one will be presented. Then, the basic functionalities of the application will be explained.

### 3.2.1 Users

In an IF, it is possible to recognize three types of user roles. The *designer* of an IF and the standard users can be separated between the *front-end user* and the *back-end user*. Each of these three roles has a different use of the system, so each has its requirements.

The designer is the user who will design and build the form. By using an IF building tool, the designer will be able to select the different structured and unstructured fields required for a form, as well as the models that will allow the validation of information, the autocompletion of structured fields, and the verification of inconsistencies between fields.

A front-end user will use the form to formalize a contact. It will have the ability to fill in the different available fields with the necessary information, as well as receive feedback and assistance to fill the form out correctly.

The back-end user is someone from the institution who will receive the contacts and will proceed with their processing. In addition, this user will be able to select the level of precision with which the models should evaluate information in the fields.

### 3.2.2 Product perspective and functions

The IF concept consists of a web application allowing a person (the front-end user) to contact an institution. This application would empower the user with tools that help fill in structured and unstructured fields more assertively and with less noise, promoting the proper expression of the intent of the interaction with the target institution. The intended features of the IF concept include:

- *Classify the information provided by the front-end user*: the use of classification models in unstructured fields will allow checking and validating whether the content is in accordance with the institution's specific domain. This functionality will enable understanding if it is possible to process the user's contact after its submission (either automatically or by the back-end users). The user will also be able to understand if the content provided meets what is expected and contains the necessary information for its proper processing. However, it will be challenging to train the models themselves if they do not have enough data. For

this reason, it will be helpful to provide a range of more general models that meet what is expected to be classified on the IF, allowing the institution to select the model most useful.

- *Verify the existence of inconsistencies between fields*: once again, when using classification models complementing with restrictions, it will be possible to verify inconsistencies and contradictoriness between structured and unstructured fields.

- *Automatic fill of structured fields*: the use of these models will allow the extraction of information from unstructured fields to fill in structured fields. Following this line of thought, the application must provide the user with feedback regarding the content provided in unstructured fields. With this, the form will be able to evaluate and complete specific structured fields. If these fields are not in accordance with what is intended by the user, the user will not be able to change them directly, having to rewrite the text provided.

- *Tools to assist in writing*: this functionality will help the user fill in the CF so that the usability and the quality of the written text are the best possible. This assistance will be possible using writing assistance tools, be they predictive text, auto-complete, and grammar/spell checkers.

The different functionalities of the application, applied to the various fields of an IF, are shown in Figure 3.1.

## 3.3 Specific requirements

This section contains in detail all the IF's functional requirements. Additionally, it provides a detailed description of the application and all its essential features.

### 3.3.1 Interface requirements

This subsection provides a detailed description of all inputs and outputs from the application. It also describes the software interfaces and provides basic user interface prototypes.

#### 3.3.1.1 User interfaces

Forms are the primary means of user interaction with an institution's website and where the main usability problems usually occur. The interfaces of an IF have common aspects, regardless of the final purpose. An IF interface must be simple, presenting what a user needs. In addition, it must maintain consistency in traversing the different interfaces it may have, using the knowledge and perception of what users can do and how they can do it.

Front-end users must clearly distinguish the fields in which completion is mandatory from the other fields. Error messages must be clear and must help the user to correct or overcome errors. These must be next to the field to which they relate. Furthermore, the user must receive feedback that their action was carried out successfully. This feedback should not be intrusive.

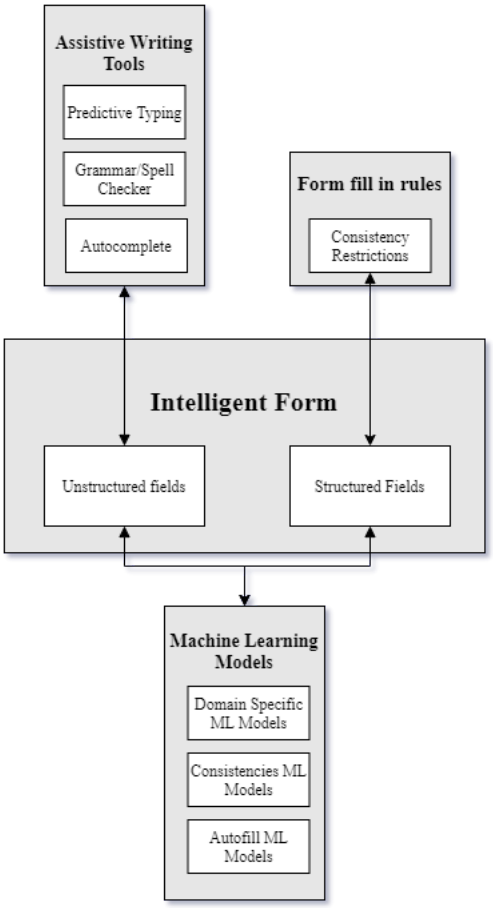


Figure 3.1: Features and functionalities applied to an Intelligent Form

The interfaces used depend on the institution in which the form will be applied. Regardless, a form is usually made up of structured and unstructured fields. Furthermore, individual institutions may require the user to log in beforehand so that it is possible to formalize the contact. In the interest of detail, we will start using the Electronic Complaints Book (ECB) as a more general case study and the ASAE's contact form as a more specific case study.

#### 3.3.1.1.1 Electronic Complaints Book

The ECB is a public online form and web app that can be used to submit complaints, compliments, suggestions, or information requests. These contacts may target any economic operator working in Portugal. This contact form is relatively permissive, so it is essential that its intelligent version helps the user to fill in structured fields.

The developed form presented below recreates the original ECB in a simplified way, that is, with fewer fields to fill in, improving the responsiveness and intelligence of the form. First, a user who accesses the form must select the intention of the contact, between formalizing a complaint or making a compliment, as seen in Figure 3.2. Then, the user can fill in identification fields, as shown in Figure 3.3.

In Figure 3.4, it is possible to visualize the UF, where the user must complete the complaint/compliment to be made. The text field should be filled out as detailed as possible so that the classification models applied can assess the information, recognize whether it is indeed a complaint/compliment, and extract the economic operator so that the fields described in Figure 3.5 are filled in automatically.

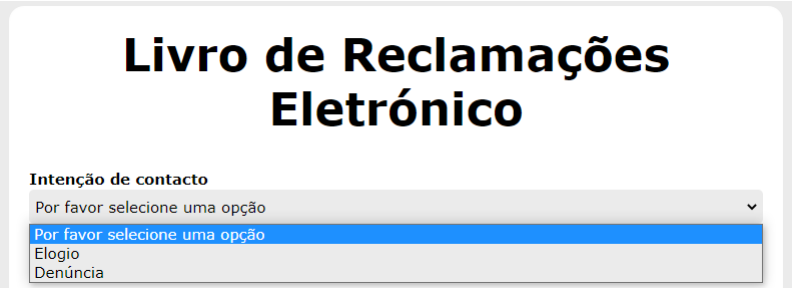


Figure 3.2: Selection field between praise and complaint.

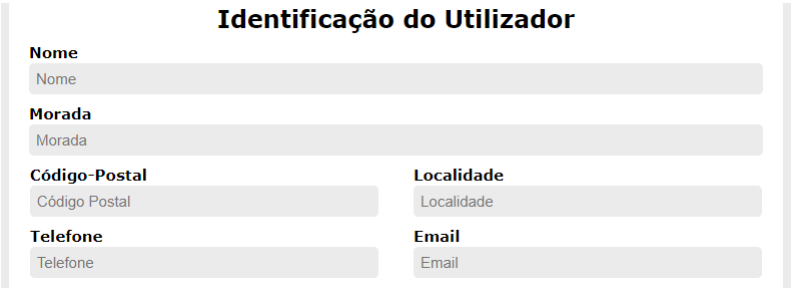


Figure 3.3: Fields for the user to identify himself.

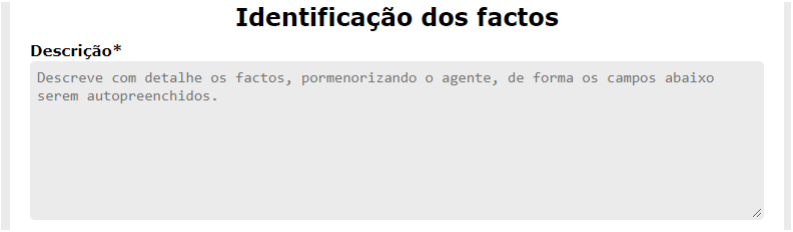


Figure 3.4: Text field that will be classified by the models.

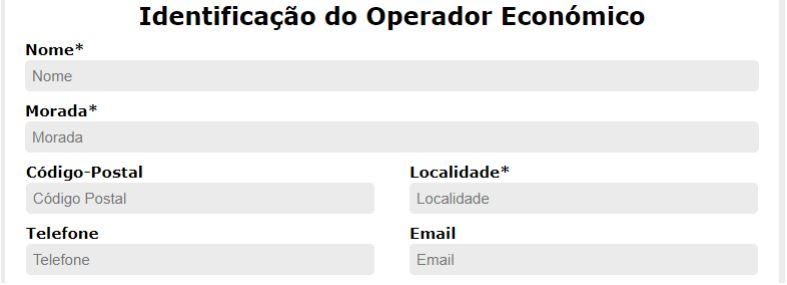


Figure 3.5: Fields automatically filled through the text field.

#### 3.3.1.1.2 Portuguese Economic and Food Safety Authority

ASAE is responsible for inspecting economic operators to assess economic and food chain hazards and enforce regulatory regulations. This administrative authority makes available on its website a form that allows formalizing complaints.

This institution has a problem with the internal processing of the complaints it receives. It is due to the fact that many complaints do not have enough information or refer to issues that are not within the jurisdiction of ASAE. For these reasons, the IF must be able to prevent these contacts from being submitted. For this, the form must be able to classify the content of the complaint in terms of economic activity and competence.

The developed form shown here is a simplified version of the original form, making the form more responsive and intelligent using fewer structured and unstructured fields. Similar to the ECB, the user has at his disposal fields that allow identifying both the user itself and the targeted economic operator.

In the ASAE form, the user must describe the event that led to the complaint. For that, it is necessary to fill in the text field with enough detail so that the classification model applied is capable of recognizing the economic activity and whether or not it is within the competence of ASAE, as seen in Figure 3.6. This ensures that the contact can be appropriately processed in the back-end.

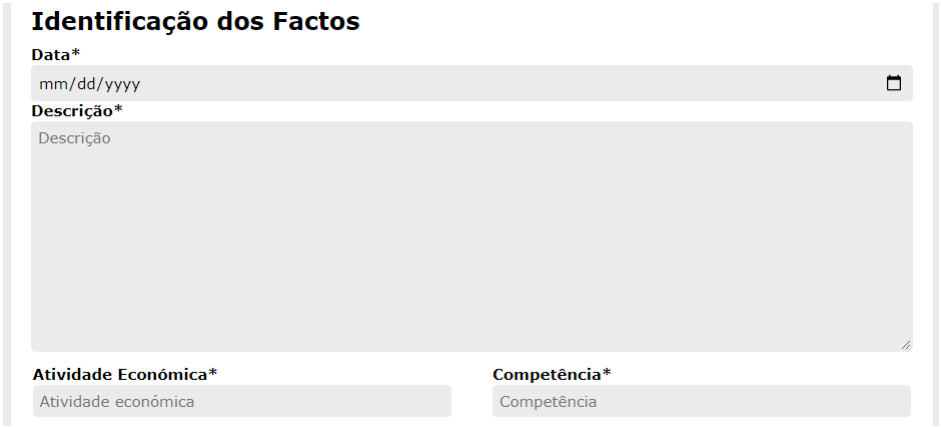


Figure 3.6: Identification of the complaint and the different key dimensions.

#### 3.3.1.2 Software interfaces

The IF needs access to APIs that allow communication with the classification models. This communication is essential for the verification of inconsistencies, auto-completion, and text classification.

### 3.3.2 Functional requirements

This section includes the requirements that specify the fundamental actions of the software system for each of the users.

#### 3.3.2.1 Designer User Case

##### 3.3.2.1.1 Functional requirement 1.1

**ID**: FR1

**Title**: Design an IF.

**Description**: A designer must be able to freely design and build a form through a simple drag & drop system.

##### 3.3.2.1.2 Functional requirement 1.2

**ID**: FR2

**Title**: Select classification models.

**Description**: In order to apply a text classification model, the designer must be able to select a model from a set of available ready-to-use models.

**Scenario**: Select the model to evaluate the content of an unstructured field.

The designer will be able to select the most appropriate classification model to classify an unstructured field.

**Scenario**: Select the model to check inconsistencies between fields.

The designer will be able to select the most appropriate classification model to check for inconsistencies between structured and unstructured fields.

**Scenario**: Select the model to allow a structured field to be automatically filled.

The designer will be able to select the most appropriate classification model to allow a structured field to be automatically filled based on the content of an unstructured field.

#### 3.3.2.2 Frontend User Case

##### 3.3.2.2.1 Functional requirement 1.3

**ID**: FR3

**Title**: Classify unstructured field.

**Description**: An IF must be able to classify textual content of unstructured fields in order to validate information according to ML text classification models, which may be specific to the target institution.

#### 3.3.2.2.2 Functional requirement 1.4

**ID**: FR4

**Title**: Check for inter-field consistency.

**Description**: An IF must be able to verify that, throughout the IF, the user has filled in the information correctly so that there are no inconsistencies between fields.

#### 3.3.2.2.3 Functional requirement 1.5

**ID**: FR5

**Title**: Automatic completion of structured fields.

**Description**: An IF must be able to automatically extract relevant information from an unstructured field and use it to fill in corresponding structured fields.

#### 3.3.2.2.4 Functional requirement 1.6

**ID**: FR6

**Title**: Text suggestions through predictive typing and auto-complete.

**Description**: An IF should be able to provide word/phrase suggestions to a user while filling an unstructured field.

#### 3.3.2.2.5 Functional requirement 1.7

**ID**: FR7

**Title**: Spell and Grammar checking.

**Description**: An IF must be able to verify that the text is written correctly, without spelling or grammatical errors and suggest changes that are necessary to correct possible errors.

### 3.3.2.3 Back-end User Case

#### 3.3.2.3.1 Functional requirement 1.8

**ID**: FR8

**Title**: Choose the precision with which the models have to classify.

**Description**: The back-end user must choose the precision with which the models classify the different fields so that it meets what is intended by the institution. By adjusting the precision with which models should evaluate textual content, the back-end user is reducing the permissibility of the form by preventing unnecessary contact submissions.

#### 3.3.2.3.2 Functional requirement 1.9

**ID**: FR9

**Title**: Select what to do with the completed information after submitting the form.

**Description**: In order to facilitate the processing of the contact made through the form, the designer will be able to select what it wants to do with the data.

**Scenario**: Select to save the data in a file.

The designer will be able to save the form information in a file after submission.

**Scenario**: Select to save the data in a database.

The designer will be able to save the form information in a database after submission.

## 3.4 Importance of requirements

This section offers the priority order of each of the requirements. The results can be seen in Table 3.1. It is possible to see these requirements ordered by importance. The priority given to each requirement was based on the innovative aspect that it would bring to the IF and not merely technical.

| Requirement ID | Title |
|---|---|
| FR1 | Design an IF |
| FR2 | Select classification models |
| FR3 | Classify unstructured fields |
| FR4 | Check for inter-field consistency |
| FR5 | Automatic completion of structured fields |
| FR8 | Choose the precision with which the models have to classify |
| FR6 | Text suggestions through predictive typing and autocomplete |
| FR7 | Spell and Grammar checking |
| FR9 | Select what to do with the completed information after submitting the form |

Table 3.1: Requirements ordered by importance

## 3.5 Summary

This chapter sets out all the minimum requirements that an Intelligent Form must contain.

Initially, the purpose of this concept is demonstrated and what purposes it may have. Then, a more detailed description of the main features of this concept and the three primary users, designer, front-end, and back-end users, and how they can use the form are presented. Finally, the requirements of the interfaces and the functional requirements of each one of the roles are introduced.

Lastly, each functional requirement was ordered according to the level of innovation it would provide for developing the Intelligent Form.

# Chapter 4

# Solution for creating Intelligent Forms



This chapter provides an exhaustive explanation of the approach taken to the concept presented heretofore, demonstrating in Section 4.1 the tool created that allows the design of IF in a generalized way. Furthermore, the technologies used to make this possible were explained. In addition, the different features available in structured and unstructured fields are also presented, together with the models available for immediate use. Section 4.2 shows a use case, the Electronic Complaints Book. In Section 4.3, the ASAE use case is described.

## 4.1 Intelligent Forms Tool

Moving from the IF concept to a practical example can be complex, especially when it requires a collection of ML models that allow performing actions such as checking inconsistencies, completing fields automatically, and classifying textual content. However, this complexity can be overcome by creating a tool that allows users to easily create HTML forms using a drag-and-drop system that allows direct application of these models.

This solution was designed to allow users to create forms, whether they knew how to program or not, and allow them to easily apply different ML models to their structured and unstructured fields, selecting them from a pool of models.

### 4.1.1 Technologies used

In order for the development of this tool to be possible, it was necessary to use a tool or framework that allowed the construction of interfaces with a simple system. Therefore, based on a research carried out and after evaluating the pros and cons of the different tools, GrapesJS[1] was selected. Figure 4.1 demonstrates the front page of the developed tool with a blank canvas, similar to the used framework.

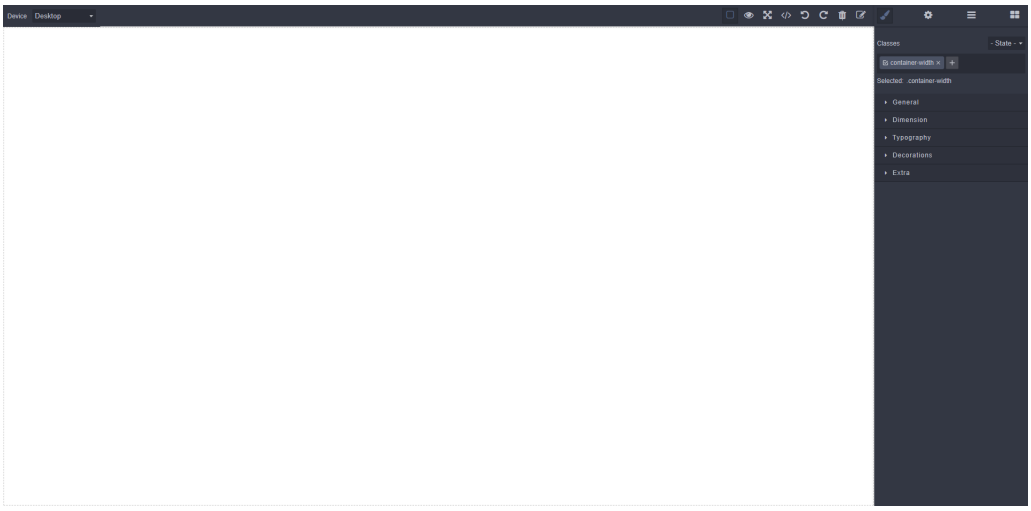


Figure 4.1: Front page of GrapesJS.

GrapesJS is an easy-to-use, open-source, visual HTML editor that makes it easy to compose HTML pages with a friendly interface. It was designed for the end-user, offering icons and compatible options so that the learning curve is minimal and productivity is even greater. In general, this framework transforms a webpage into a visual HTML and CSS editor. In addition, it allows developing templates and components using only HTML and JavaScript.

The framework used provides in its base version some basic blocks that allow the construction of these pages, as can be seen in Figure 4.2. In addition to the base version, GrapesJS provides plugins that allow the addition of features such as adding components and blocks for forms, as seen in Figure 4.3. Using the forms plugin[2] it was possible to recreate these blocks, adapting the components, providing them with new traits related to the application of ML models, and introduce some of the new features.

GrapesJS also provides additional features such as independent styling of any component inside the canvas, responsive design giving the necessary tools to optimize templates on any device, and makes the source code available so that it is easier to obtain and change, as seen in Figure 4.4.

In order to make the application of ML models possible and simplify the design, administration, and use, a study was also carried out on the best way to integrate these models in the tool. The use of models invoked through an API was the solution used.

[1] https://grapesjs.com/
[2] https://github.com/artf/grapesjs-plugin-forms

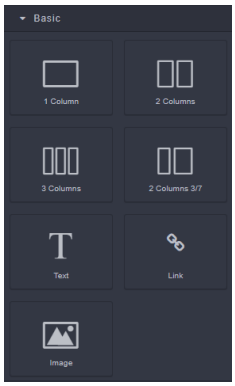


Figure 4.2: Built-in basic blocks.


Figure 4.3: Form components and blocks.

RapidAPI[3] is an API Marketplace used by over one million developers to find, test, and connect to thousands of APIs with a single account, API Key, and SDK. With this marketplace, the developer can consume any API using a unified REST format that is easy to understand and embed in an app. Moreover, the developer can view all of the APIs that are connected to using the dashboard, which monitors things like the number of API requests, latency, and error rates.

### 4.1.2 Functionalities developed

In the development of this tool, some of the features mentioned in Chapter 3 were selected so that the construction of Intelligent contact Forms was possible. The method of selection of the functionalities to be implemented was to take into account the importance and the innovative aspect that these functionalities would bring to the tool and to the objective of this work:

- *Design an Intelligent Form*.
- *Select classification models*.
- *Classify unstructured fields*.
- *Check for inter-field consistency*.
- *Automatic completion of structured fields*.

The remaining features could not be implemented due to time constraints.

[3] https://rapidapi.com/marketplace

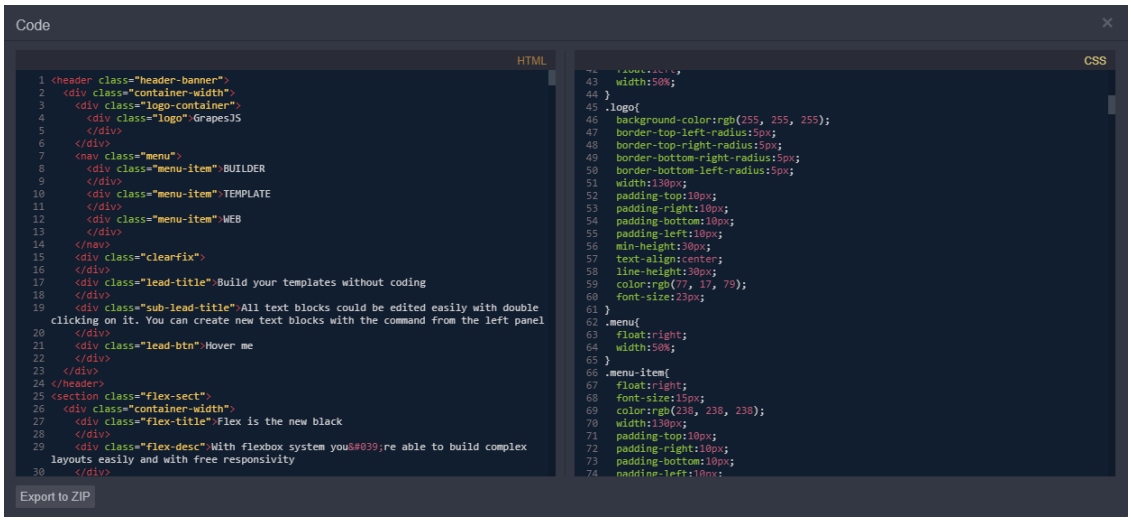


Figure 4.4: GrapesJS code editor.

### 4.1.3 Field Features

A form is composed of structured and unstructured fields. The solution involves introducing the functionalities mentioned above in these fields to improve the quality of the information provided by users, enhancing the chances of proper handling of the contact downstream. Depending on their structured or unstructured nature, form fields can be given different functionality.

Unstructured fields are the protagonists of the concept of IF that is aimed in this work. By default, an unstructured field allows a user to provide free form text: a description of the incident complained about, an exposition of an improvement suggestion, and an explanation of a critique. The most common HTML typed used in this type of field is `<textarea>`. Structured fields are fields with a fixed input or format, such as the date of the incident or incident location. Example HTML types used in this type of field are `<input>` and `<select>`.

ML models target the textual content of unstructured fields. Through this content, it is possible to apply features such as:

- *Check for inconsistencies*: the form can check if there is an inconsistency of information between a structured field filled in or selected by the user and the textual content written in an unstructured field.
- *Auto-complete/Auto-select*: the form can extract information from the content of an unstructured field in order to fill in input or automatically select an option in a select.
- *Filter information*: the form can extract information from the content of an unstructured field in order to filter options in select fields in order to reduce the number of available options.
- *Allow/disable submissions*: the form can classify the content of an unstructured field and, depending on the result and its value, allow or block its submission.

Figure 4.5 presents a schematic of how these dependencies between fields are interconnected. In short, the unstructured field contains the necessary textual content for both the classification and extraction of entities. From these actions, the form can perform the implemented functionalities using the content of the structured fields.

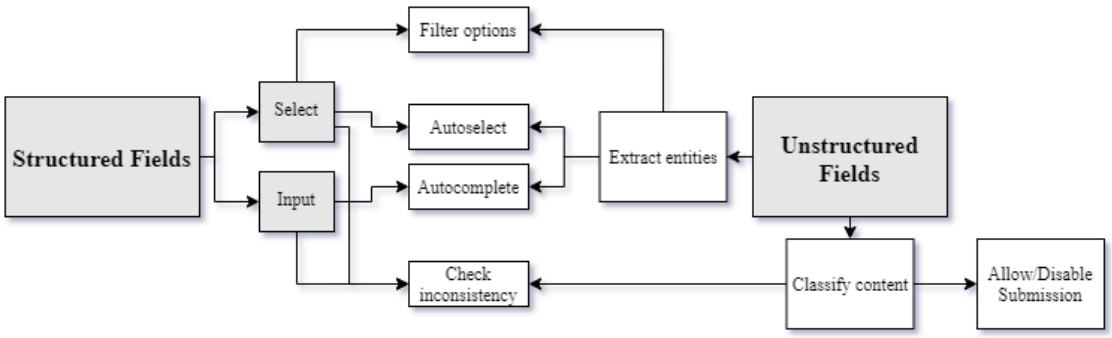


Figure 4.5: Scheme of field interdependencies in an Intelligent Form.

### 4.1.4 Machine Learning Models

Applying ML models to a form is a difficult task, especially for people with no programming skills. The idea of the tool developed is to be sufficiently easy to use to avoid the need for that kind of knowledge or background. To facilitate the application of text classification models, the tool contains a set of pre-collected models. These can be easily applied to different unstructured form fields using just the tool's interface, as seen in Figure 4.6.

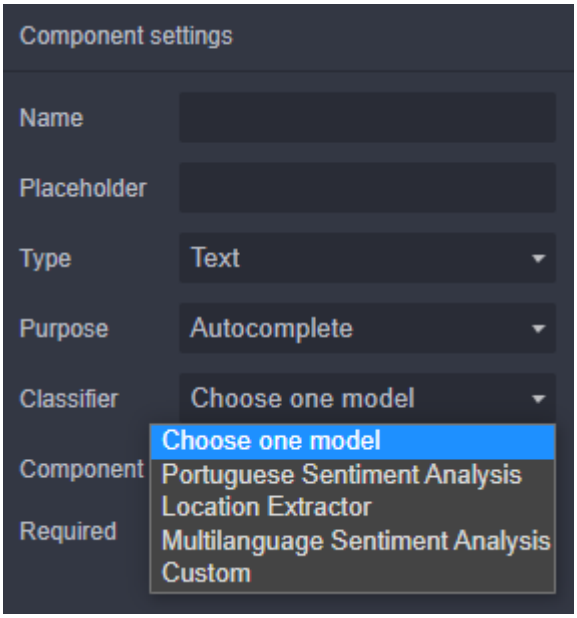


Figure 4.6: Models available in the tool for an input element.

Next, examples of classification and extraction models used in the developed tool illustrate the concept of the ready-to-use model collection.

#### 4.1.4.1 Portuguese Sentiment Analysis

The inclusion of a Sentiment Analysis model[4] was made, keeping in mind the idea of being applied to IF to detect and monitor their users' sentiment and overall opinions. The sentiment is usually negative or positive, and it is extracted from the input of the user.

#### 4.1.4.2 Multilanguage Sentiment Analysis

The inclusion of another Sentiment Analysis model[5] came from having a more diversified set and classifying texts from other languages. This model, in addition to Portuguese, accepts texts in Spanish and English. Furthermore, this model differs from the previous model in that it has a different polarity, classifying texts as neutral.

#### 4.1.4.3 Location Extractor

This model[6] is a package of NLP, Information Retrieval, and ML tools that extracts named entities (people, organizations, products, and locations) and values (URLs, emails, telephone numbers, currency amounts, and percentages) mentioned in a body of text.

#### 4.1.4.4 Custom Model

When dealing with particular cases, it is sometimes necessary to use domain-specific classification or extraction models. For this reason, the designer can make use of self-developed models that assess relevant information. When the designer selects this option, it must access the code editor that the tool provides and introduce the invocation to the API that contains the model to be used.

The following two sections present the versions of the forms created for the case studies that motivate this work. Both versions were designed using the IF build tool developed.

## 4.2 Electronic Complaints Book

The digital version of the complaints book became mandatory and transversal to all sectors of the national economy. Through electronic means, a complaint can be filed at any time.

This form was recreated more straightforwardly using the developed tool mentioned above so that the features presented in Chapter 3 can be illustrated. The idea is to collect complaints and compliments, as is possible in the original ECB[7].

This case study is intended to demonstrate that it is possible to help the user fill out the form correctly. For this, two features were applied: inconsistency check between fields and automatic completion of structured fields.

[4] https://rapidapi.com/metatext-metatext-default/api/sentiment-analysis-portuguese-br/
[5] https://rapidapi.com/logicione/api/entity-and-sentiment-extractor/
[6] https://rapidapi.com/aylien/api/text-analysis/
[7] https://www.livroreclamacoes.pt/INICIO

Figure 3.2 shows a structured field, a dropdown list with two options (Compliment and Complaint). Usually, a compliment has a positive connotation, and a complaint has a negative connotation. Therefore, the Portuguese Sentiment Analysis model was applied in the unstructured field shown in Figure 3.4 to evaluate the content and assigned it a positive or negative value, then comparing it with the dropdown options in order to verify inconsistency.

Another feature available is auto-completion through filtration. The model also uses the unstructured field in Figure 3.4 to extract all location and organization entities. Then, filtering is done with these entities in a database of 175 736 economic operators, obtained from the database that feeds the ASAE management and information system, the GESTASAE, to restrict options. In this functionality, we can get two situations:

- *Select the correct option*: the user must select from the filtered available options the operator referred to, as can be seen in Figure 4.7
- *Direct auto-completion*: the form was able to select from the database the economic operator referred to, as shown in the Figure 4.8

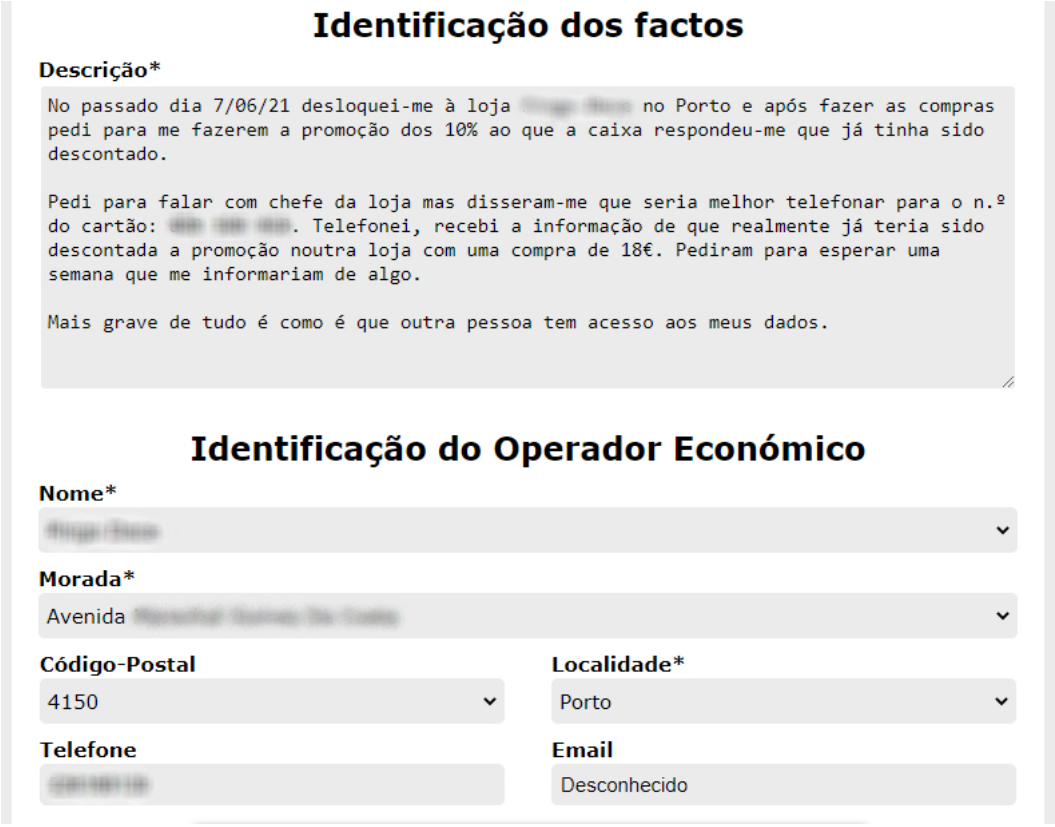


Figure 4.7: Filtering of economic operators by the location extraction model.

As mentioned, the design of this form was made to help the user correctly fill in the form. To test the effectiveness of the implemented functionalities, an evaluation was made using volunteers who used the form. It is possible to see this evaluation in Section 5.1.

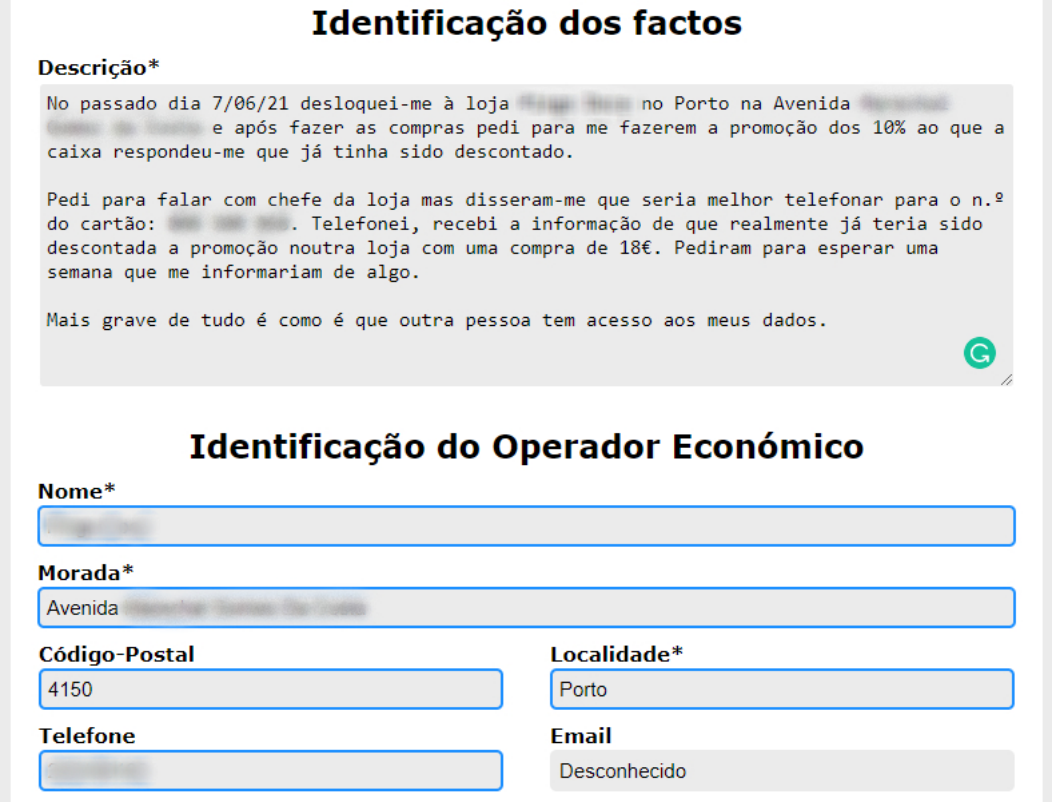


Figure 4.8: Filtering economic operators until only one result is obtained.

## 4.3 ASAE's Form

ASAE's electronic complaint form[8] is intended to facilitate the communication of unlawful facts to ASAE, which may be of an administrative or criminal nature within the competence of this authority. Similar to the ECB form, a version of the form was recreated.

In Figure 3.6 it is possible to visualize three essential fields. The unstructured field will be the target of ML models, that is, the input the models will use to classify the complaint. In this field, two machine learning models are employed, which evaluate textual content in two dimensions. The first dimension is the type of economic activity identified in the complaint. The second key dimension determines whether the subject enclosed in a complaint is within the competence of ASAE.

When the text is evaluated in these two dimensions, the fields at the bottom of the Figure 3.6 are filled in, making the user making the user aware in advance of how the complaint filled out was evaluated. After classifying the content of the complaint in the two dimensions mentioned, the form proceeds to evaluate and determine what the user can do:

- *Competence*: it is only possible to submit the form if the complaint is determined to be within the competence of ASAE.

[8] https://www.asae.gov.pt/queixas-e-denuncias.aspx

- *Economic Activity*: the form can identify one or more economic activities of the complaint made among the ten possible classes. If no activity is identified with a minimum certainty value, the form cannot be submitted.

The models used must classify in these two dimensions with a level of precision superior to the threshold predefined by the backend user. Blocking the form submission encourages users to write a complaint with consistent and complete information.

The economic activity classification model consists of 10 binary classifiers that provide the probability of the complaint belonging to each of the economic activities: *Primary production*, *Industry*, *Restauration*, *Wholesalers*, *Retail*, *Direct selling*, *Distance selling*, *Production & trade*, *Service Providers* and *Safety & environment*

The competence classification model is also a binary classifier that indicates the probability that the assessed complaint is and is not within the competence of the ASAE.

Given that the aim is to minimize the number of complaints that are not in ASAE's interest submitted, thus improving the quality of the complaints received by ASAE, only economic activities that have a probability above the threshold and that are within the competence of ASAE with a probability above the threshold are considered valid. The backend user then receives the complaint with all these activities sorted by probability and selects the most suitable one.

As with the ECB, an evaluation of the efficiency of the developed form was also carried out. In this case, a dataset was used to perform metric evaluations. It is possible to see this experiment in more detail in Section 5.2.

## 4.4 Summary

The discovery of the presented solution came from the requirements gathered in Chapter 3. The tool presented was developed thinking about a simple and easy way to build forms, providing ready-to-use machine learning models to people who do not have knowledge in programming.

Initially, an examination of the general problems of the contact form of Electronic Complaints Book and ASAE was carried out. From these problems, two instances were made to determine if the development of an Intelligent Form would resolve these same problems. Based on the requirements and the two instances, it was feasible to develop a tool that would enable the construction of an Intelligent Form with functionalities that would support and help the user fill in the form correctly, providing a quality contact.

Finally, from this same tool developed, the two forms previously analyzed were elaborated. In this way, it was possible to implement functionalities that allow us to assist in filling out forms and evaluating their content beforehand, thus allowing institutions to process them more quickly and without the need to collect new information.

In the next chapter, two experiments carried out on these two Intelligent Forms are presented, developed in order to validate their efficiency and the implemented functionalities.

# Chapter 5

# Experimental Evaluation



This chapter contains a description of the experiments conducted to test the proposed methodology for both case studies. Also, in this chapter, the experimental results are revealed, which are later analyzed and discussed. In Section 5.1 the Electronic Complaints Book case study experiment is presented. Then, it is possible to visualize the results obtained from the experiment carried out, namely the type of content filled with the respective result and the problems found. Section 5.2 covers the various details of the dataset used and the description of the experimental procedures aimed at testing the solution developed for the ASAE case study. Below, the results of the experiment accomplished are presented, with a performance evaluation being carried out.

## 5.1 User Experience Evaluation

A comprehensive assessment of an interactive system demands human judgment. User experience must be evaluated and analyzed. This step of validation is crucial to improve the system and its functionality to fulfill user requirements.

### 5.1.1 Experimental setup

In order to properly study the user experience dimension of the IF, a usability test was designed. This type of study requires the presence of volunteers to interact with the IF.

The first stage of this assessment focuses on gathering volunteers who have already been using contact forms and who have in some way made a complaint or compliment to an economic operator. Furthermore, it is critical to ensure that volunteers understand the purpose of the task they are

given. For this, they were given a brief explanation of the ECB contact form's problems and asked to submit a compliment or complaint in the IF provided. Then, the volunteer should access the form, select the intention of contact, fill in the fields referring to the user's identification and fill in the unstructured field referring to the description of the contact. Participants could ask questions while performing the task.

The participant should understand from the indications shown in the form that subsequent fields would be automatically filled out. Suppose the form could not automatically fill in the fields relating to the economic operator. In that case, two situations could occur: dropdowns with the various options relating to the economic operator were made available to the user to select, or the user would have to fill in the fields manually.

The only indications the participants received were that they should treat the exercise of filling the form as a real case. That is, the situation might or might not be accurate, but the economic operator would have to be authentic.

The next phase of this assessment is to ask participants to complete a survey to understand the final result of the task performed. Depending on the final result of the contact, the user had to select what occurred while filling out the contact.

### 5.1.2 Results

This section presents the results obtained from the experiment concerning the user's experience with the ECB form. The test was carried out by a total of twelve volunteers chosen for convenience. Half of the participants have been assigned a compliment contact intention, while the other half have been told to make a complaint. All participants understood what was intended to be accomplished with the given task. The volunteers have indicated the problems they encountered when filling out the form.

This IF intends to assist the user in filling out structured fields by extracting entities from the information filled in the unstructured field. In addition, the form must evaluate the content of that same field and validate if it is consistent with the selected contact intent (compliment or complaint).

The filtering of economic operators depends on the user who fills out the form, as it may or may not detail the operator sufficiently, and it also depends on the type of operator. For example, if the user makes a description referring to an operator with several establishments, the inserted text must be as detailed as possible. However, the same does not happen when the user refers to a very explicit operator.

Of the twelve participants in this experiment, all were able to submit their contact details. This means that the Sentiment Analysis model used to classify the intention of the contact was successful in these contacts. For a contact to be possible to submit, the textual content must be clear regarding the intention of contact and detailing the economic operator. The following example is a text from one of the participants and demonstrates a successful contact:

> "No passado domingo realizei um almoço de família no restaurante Bagoeira em Barcelos. A comida estava divinal e os funcionários trouxeram as bebidas tal como pedidas, extremamente frescas. O ambiente estava também fresco, permitindo uma harmonia e um belo almoço."

The volunteer gave a compliment, clearly demonstrating a positive feeling for the economic operator, detailing its name and location.

Of the participants who successfully submitted their contact details, four had to fill manually in the details regarding the economic operator, despite having indicated the name or location of the economic operator. This is due to two reasons: the NER model could not extract entities to filter economic operators, or the algorithm used for filtering could not reduce the initial number of economic operators with the provided entities. The following example is a text from one of the participants and demonstrates a contact in which the participant had to fill manually in the economic operator's fields:

> "No dia 17 de junho de 2021 fui ao restaurante The Green Affair. Apesar de ser um local muito concorrido, consegui arranjar mesa sem reserva. O espaço é agradável e a decoração é muito bonita. Quanto à comida, achei incrivelmente apetitosa e com uma boa relação preço-qualidade. Apesar disso, aquilo que mais gostei foi o atendimento. Todos foram muito simpáticos e atenciosos. Para repetir. "

The NER model could not extract entities in the example presented, forcing the participant to fill in the fields manually. This demonstrates that part of the objective of this IF has not been achieved in this specific case.

Three of the four participants who had to fill manually in the economic operator were due to the algorithm fails. This is justified either because the NER did not extract the correct entities or that the database does not contain economic operators that correspond to the entities mentioned.

In addition to these four participants who had to fill manually in the economic operator's details, another four, despite the form being able to extract and filter the number of economic operators, chose to manually fill in these fields as the economic operator was unavailable in the options provided. The problem is caused by the database that has only 175 736 economic operators. This problem is easily solved using a complete database and with a more significant number of operators.

On the other hand, four participants were assisted to fill in the economic operator's fields. Of these, three had to select the referred operator among the options, and only one had the fields automatically filled out. Table 5.1 details the number of options provided to the participants and the number of entities mentioned extracted.

| Participant | Number of economic operators | Number of entities extracted |
|---|---|---|
| **#1** | 1 | 3 |
| **#2** | 266 | 1 |
| **#3** | 7 | 1 |
| **#4** | 2 | 2 |

Table 5.1: Number of economic operators filtered against the number of entities mentioned extracted

As it can be observed, the number of entities is not directly linked to the number of options provided. However, the more detailed the user is, the smaller the number of options is since there is significant filtering.

### 5.1.3 Discussion

The Electronic Complaints Book IF was developed to assist users in filling in the form. Using classification models, it is possible to inform users of any problems that their contact may have. Furthermore, these same models make it possible to prevent the submission of low-quality contacts.

Through this evaluation carried out with the help of volunteers, it was possible to conclude that the form developed to demonstrate the IF concept effectively prevents the submission of contacts that do not have congruent information. Furthermore, the use of a NER model was able to assist users in filling in the economic operator's fields.

However, it should be noted that generic models (of Sentiment Analysis and NER) are used and not trained in data from this domain (complaints and compliments about economic operating operators in Portugal). Therefore, this form needs three significant updates: the Sentiment Analysis model trained in data of compliments and complaints, a better performing NER model, and a more complete database.

An assessment with more participants can also be carried out to understand any usability issues.

## 5.2 Performance Evaluation

ASAE is one of the target institutions of study and motivation for carrying out this work, as mentioned before. For this reason, it is sensible to evaluate the performance of the solution built to reduce the submission of poor-quality complaints. For that, we inject real cases of complaints in the IF.

### 5.2.1 Dataset

The dataset used includes 85 950 entries of complaints received via the ECB between July 2018 and April 2021. Each entry in the dataset contains an identification number, the complaint content, and the respective economic activity and competence. Notwithstanding, some of the entries are

not labeled. For this reason, these entries could not be considered for evaluation purposes, as there would not be an element of comparison. Such a problem leads to the urge to clean the data, leading to a final number of 20 660 complaints. It is important to note that this dataset of complaints from the ECB was not used to train the models applied in the form.

As mentioned above, a complaint can belong to one of the ten economic activities. The data distribution by economic activity dimension may be found in Figure 5.1. Regarding the determination of the competent authority to handle a complaint, some complaints have shared responsibilities between ASAE and other entities. In Figure 5.2, it is possible to observe the distribution of complaints according to the dimension of competence. It is possible to observe this distribution in more detail in Tables A.1 and A.2 provided in Appendix B.

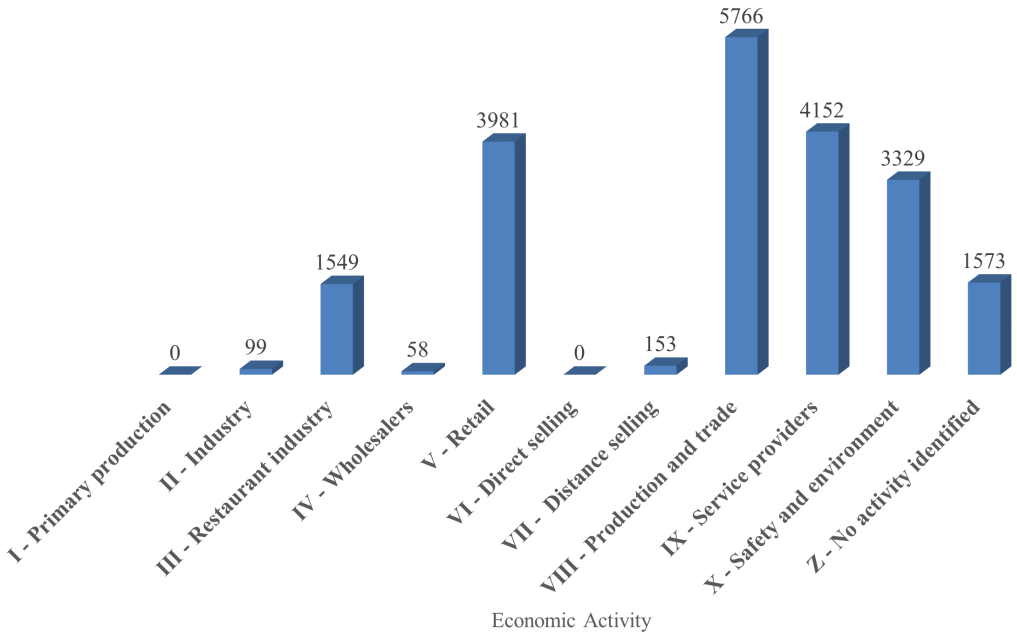


Figure 5.1: Distribution of complaints by economic activity in the dataset.

This dataset was then used to evaluate the intelligent version of the ASAE'S form developed and presented in Section 4.3. Given the two classification models used in this form to validate the introduced data related to the complaint's two dimensions, the form works as a binary classification model: either it is possible to submit the complaint or not.

### 5.2.2 Experimental setup

Puppeteer is a Node library that provides a high-level API to control Chrome without a visible UI shell, allowing test automation in modern web applications. Through the use of this library, it was possible to inject the 20 660 complaints of the dataset into the IF

Before starting the experiment, it was necessary to calculate the expected result of each of the complaints if inserted into the unstructured field of the developed ASAE form. For this, the two

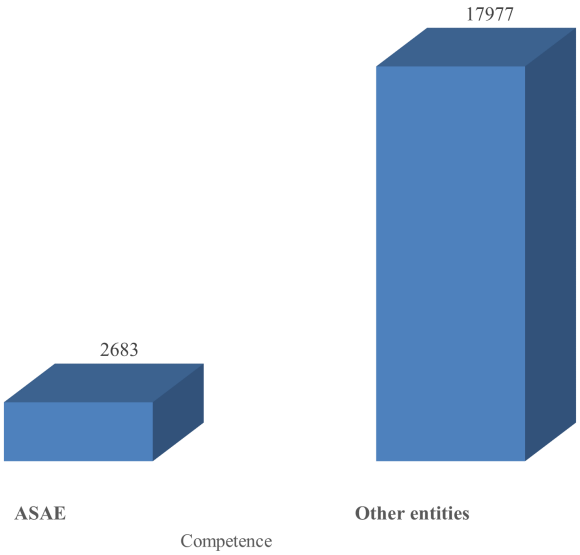


Figure 5.2: Distribution of complaints by competence in the dataset.

key dimensions available in the dataset and previously manually labeled were used to calculate whether the complaint could be submitted. Complaints that cannot be submitted are those that are not within the competence of ASAE or that do not have an identified economic activity. The result of this calculation is shown in Figure 5.3.

We have developed a script that deals with reading the necessary data from the dataset and injecting it into the ASAE form. When the models used in the form finish classifying the complaint, the two dimensions and respective probability are collected and whether or not it is possible to submit the form. Due to the high number of complaints, it was necessary to resort to asynchronous processing in order to be able to inject several complaints at the same time.

After injecting the 20 660 complaints on the developed form, a performance analysis was performed by changing the thresholds and evaluating the form on several evaluation metrics, such as Accuracy, Precision, Recall, and F1 Score.

### 5.2.3 Results

This section presents the results obtained from the experiment concerning the performance of the ASAE form. As explained previously, the experiments in this research work intended to ultimately verify if it was possible to improve the quality of the contacts submitted to the institutions. Thus, during the experiments, it is expected that, in the end, it will be possible to check different numbers of complaints submitted and not submitted.

As the experiments' process is already executed with the dataset presented above, one of the most important aspects is the performance. Let us make the following definitions:

- "Can be submitted" is a Positive class.

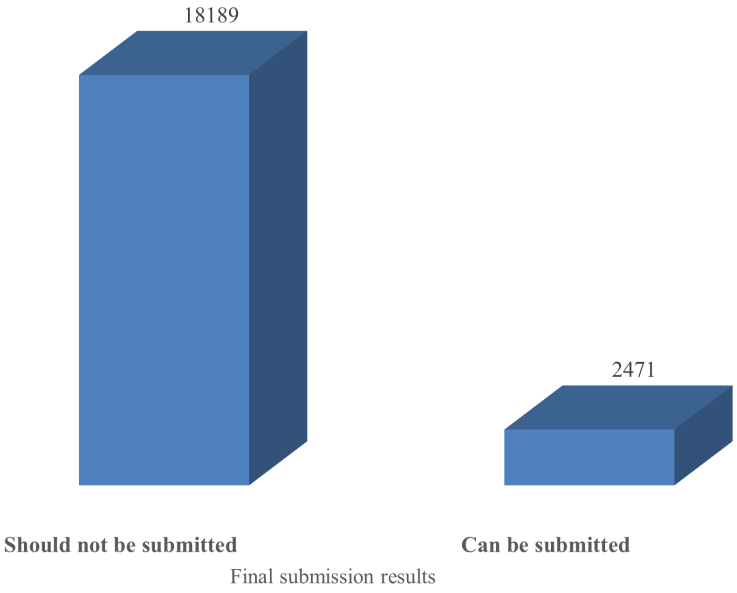


Figure 5.3: Distribution of complaints by possibility of submitting.

- "Cannot be submitted" is a Negative class.

The "Submission-prediction" can be summarized using a 2x2 confusion matrix that depicts all four possible outcomes, as seen in Table 2.1.

Therefore, this section aims to exhibit the different performance levels, with the experiments conducted, using different thresholds. The *Accuracy*, *Precision*, *Recall*, and *F1* scores will be computed to assess performance.

#### 5.2.3.1 Confusion matrix and evaluation metrics

Figure B.1 and B.2 in Appendix B illustrate the distribution of the outcomes of the 20 660 complaints after the form has examined them. In Table B.1 and Table B.2, it is possible to observe the confusion matrix built with the results obtained. Despite quickly visualizing a discrepancy between actual values and predicted values, the objective of this work is not to assess the quality of the models used but to assess the impact that their use has on the possible submission of complaints and how this can prevent the submission of poor quality contacts.

As a result, it was possible to calculate the number of correct and incorrect predictions, thus creating a confusion matrix, as shown in Table 5.2.

| | | Predicted values | |
|---|---|---|---|
| | | Positive (P) | Negative (N) |
| **Actual values** | Positive (P) | 1 072 | 1 521 |
| | Negative (N) | 4 864 | 13 203 |

Table 5.2: Confusion matrix of experiment with threshold 0.5 for both tasks

Through these collected values and the constructed confusion matrix, it is possible to calculate evaluation metrics for the classification. It is possible to recalculate the results obtained by varying the economic activity threshold value and the competence threshold value.

The results obtained through these variations for Accuracy, Precision, Recall, and F1 score are available below. Bearing in mind that the main objective is to reduce the submission of poor-quality complaints, especially complaints that are not within the competence of ASAE, the values of these metrics must be balanced.

The results can be interpreted in different ways. The accuracy values are available in Table 5.3. In order to reduce the number of contacts that are not within the competence of the ASAE, a threshold for competence above 0.7 indicates an adequate level of precision, mainly for high threshold values of economic activity, as can be seen in Table 5.4. However, for these values, the Recall value is low (see Table 5.5), which indicates that there is a loss of important contacts.

| | | Economic Activity Threshold | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | **Accuracy** | **0.1** | **0.2** | **0.3** | **0.4** | **0.5** | **0.6** | **0.7** | **0.8** | **0.9** |
| **Competence Threshold** | **0.1** | 0.6911 | 0.6911 | 0.6911 | 0.6998 | 0.8010 | 0.8572 | 0.8682 | 0.8727 | 0.8744 |
| | **0.2** | 0.6911 | 0.6911 | 0.6911 | 0.6998 | 0.8010 | 0.8572 | 0.8682 | 0.8727 | 0.8744 |
| | **0.3** | 0.6911 | 0.6911 | 0.6911 | 0.6998 | 0.8010 | 0.8572 | 0.8682 | 0.8727 | 0.8744 |
| | **0.4** | 0.6911 | 0.6911 | 0.6911 | 0.6998 | 0.8010 | 0.8572 | 0.8682 | 0.8727 | 0.8744 |
| | **0.5** | 0.6911 | 0.6911 | 0.6911 | 0.6998 | 0.8010 | 0.8572 | 0.8682 | 0.8727 | 0.8744 |
| | **0.6** | 0.8565 | 0.8565 | 0.8565 | 0.8575 | 0.8642 | 0.8705 | 0.8725 | 0.8738 | 0.8745 |
| | **0.7** | 0.8742 | 0.8742 | 0.8742 | 0.8742 | 0.8741 | 0.8739 | 0.8742 | 0.8746 | 0.8746 |
| | **0.8** | 0.8746 | 0.8746 | 0.8746 | 0.8746 | 0.8746 | 0.8745 | 0.8745 | 0.8745 | 0.8745 |
| | **0.9** | 0.8745 | 0.8745 | 0.8745 | 0.8745 | 0.8745 | 0.8745 | 0.8745 | 0.8745 | 0.8745 |

Table 5.3: Accuracy result varying threshold values for economic activity and competence

| | | Economic Activity Threshold | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | **Precision** | **0.1** | **0.2** | **0.3** | **0.4** | **0.5** | **0.6** | **0.7** | **0.8** | **0.9** |
| **Competence Threshold** | **0.1** | 0.1807 | 0.1807 | 0.1807 | 0.1784 | 0.1926 | 0.2468 | 0.2575 | 0.2564 | 0.4444 |
| | **0.2** | 0.1807 | 0.1807 | 0.1807 | 0.1784 | 0.1926 | 0.2468 | 0.2575 | 0.2564 | 0.4444 |
| | **0.3** | 0.1807 | 0.1807 | 0.1807 | 0.1784 | 0.1926 | 0.2468 | 0.2575 | 0.2564 | 0.4444 |
| | **0.4** | 0.1807 | 0.1807 | 0.1807 | 0.1784 | 0.1926 | 0.2468 | 0.2575 | 0.2564 | 0.4444 |
| | **0.5** | 0.1807 | 0.1807 | 0.1807 | 0.1784 | 0.1926 | 0.2468 | 0.2575 | 0.2564 | 0.4444 |
| | **0.6** | 0.2812 | 0.2812 | 0.2812 | 0.2838 | 0.2924 | 0.3139 | 0.2887 | 0.3158 | 0.5000 |
| | **0.7** | 0.4685 | 0.4685 | 0.4685 | 0.4673 | 0.4512 | 0.3696 | 0.3704 | 0.6154 | 1.0000 |
| | **0.8** | 0.6000 | 0.6000 | 0.6000 | 0.6000 | 0.5714 | 0.5455 | 0.5000 | 0.5000 | 1.0000 |
| | **0.9** | X | X | X | X | X | X | X | X | X |

Table 5.4: Precision result varying threshold values for economic activity and competence

X means the value is impossible to calculate. This is due to the number of True Positives and False Positives being zero.

| | | Economic Activity Threshold | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Recall | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 | 0.6 | 0.7 | 0.8 | 0.9 |
| Competence Threshold | 0.1 | 0.4134 | 0.4134 | 0.4134 | 0.3860 | 0.1836 | 0.0671 | 0.0266 | 0.0077 | 0.0015 |
| | 0.2 | 0.4134 | 0.4134 | 0.4134 | 0.3860 | 0.1836 | 0.0671 | 0.0266 | 0.0077 | 0.0015 |
| | 0.3 | 0.4134 | 0.4134 | 0.4134 | 0.3860 | 0.1836 | 0.0671 | 0.0266 | 0.0077 | 0.0015 |
| | 0.4 | 0.4134 | 0.4134 | 0.4134 | 0.3860 | 0.1836 | 0.0671 | 0.0266 | 0.0077 | 0.0015 |
| | 0.5 | 0.4134 | 0.4134 | 0.4134 | 0.3860 | 0.1836 | 0.0671 | 0.0266 | 0.0077 | 0.0015 |
| | 0.6 | 0.0922 | 0.0922 | 0.0922 | 0.0891 | 0.0578 | 0.0270 | 0.0108 | 0.0046 | 0.0012 |
| | 0.7 | 0.0201 | 0.0201 | 0.0201 | 0.0193 | 0.0143 | 0.0066 | 0.0039 | 0.0031 | 0.0012 |
| | 0.8 | 0.0035 | 0.0035 | 0.0035 | 0.0035 | 0.0031 | 0.0023 | 0.0015 | 0.0008 | 0.0004 |
| | 0.9 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |

Table 5.5: Recall result varying threshold values for economic activity and competence

Although the objective is to reduce the entry of complaints that are not within the ASAE's competence, it is also essential to bear in mind that it is necessary to pre-process these contacts and, for these reasons, it is preferable to have a balance of values. Therefore, looking at Table 5.6, values lower than 0.5 for competence and 0.4 for economic activity indicate higher values than the others.

| | | Economic Activity Threshold | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | F1 | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 | 0.6 | 0.7 | 0.8 | 0.9 |
| Competence Threshold | 0.1 | 0.2515 | 0.2515 | 0.2515 | 0.2440 | 0.1880 | 0.1055 | 0.0482 | 0.0150 | 0.0031 |
| | 0.2 | 0.2515 | 0.2515 | 0.2515 | 0.2440 | 0.1880 | 0.1055 | 0.0482 | 0.0150 | 0.0031 |
| | 0.3 | 0.2515 | 0.2515 | 0.2515 | 0.2440 | 0.1880 | 0.1055 | 0.0482 | 0.0150 | 0.0031 |
| | 0.4 | 0.2515 | 0.2515 | 0.2515 | 0.2440 | 0.1880 | 0.1055 | 0.0482 | 0.0150 | 0.0031 |
| | 0.5 | 0.2515 | 0.2515 | 0.2515 | 0.2440 | 0.1880 | 0.1055 | 0.0482 | 0.0150 | 0.0031 |
| | 0.6 | 0.1388 | 0.1388 | 0.1388 | 0.1356 | 0.0966 | 0.0497 | 0.0208 | 0.0091 | 0.0023 |
| | 0.7 | 0.0385 | 0.0385 | 0.0385 | 0.0370 | 0.0277 | 0.0129 | 0.0076 | 0.0061 | 0.0023 |
| | 0.8 | 0.0069 | 0.0069 | 0.0069 | 0.0069 | 0.0061 | 0.0046 | 0.0031 | 0.0015 | 0.0008 |
| | 0.9 | Y | Y | Y | Y | Y | Y | Y | Y | Y |

Table 5.6: F1 Score result varying threshold values for economic activity and competence

Y means the value is impossible to calculate. This is due to the value of Precision and Recall being zero.

### 5.2.4 Discussion

As far as we know, this project presents a new approach to CF using classification models to improve communication between users and institutions, helping those who fill in the forms. The developed ASAE form works as a binary classification model, meaning two classes work as true and false. In this specific case, we have the possibility of submission or not.

A critical step in obtaining an optimized classifier is selecting an appropriate measure for discriminating the optimal solution.

In the detection of submission of complaints, a false positive means that a complaint that cannot be submitted has been identified as possible to submit. The institution receives complaints with incomplete information, which is not within its competence or whose activity is incorrect, leading to this form not fulfilling its purpose. This means that it needs high precision values for the form to prevent the entry of unnecessary contacts.

On the other hand, the level of recall also needs to be satisfactory. Since we deal with complaints from economic operators that sometimes commit serious infringements, it is expected that there will be no loss of information relevant to the institution.

Since we are dealing with two classification models with different threshold levels, and it is necessary to discard complaints that are not in the institution's interest without preventing the entry of essential complaints, it is necessary to find a balance between the two. It is necessary to find a high F1 score since we are looking for the balance between Accuracy and Recall. Furthermore, we are dealing with a dataset where the distribution is uneven, with a high number of True Negatives.

Finally, this experiment demonstrates that there are still improvements to be implemented in the IF concept in order to be possible to achieve better performance levels regarding the submission of better quality complaints.

## 5.3 Summary

This chapter explores the usability and user experience tests crucial for a thorough evaluation of the Electronic Complaints Book form. In addition, the results of the experiments carried out using the ASAE form are also presented. Finally, the statistical analysis of the results obtained is also presented and discussed in this chapter.

The first experiment was carried out in order to understand how Intelligent Forms influence the user experience. For this, a direct evaluation of the Electronic Complaints Book intelligent form using volunteers was planned. After completing the task, each participant was asked to respond to a survey to understand any problems. With the results obtained, it is possible to conclude that the IF can verify inconsistencies between fields and that it helps the user to fill in structured fields. However, the user must write with a certain level of detail so that the NER model can extract the entities. In addition, there are some improvements to be made in this form so that its performance is even better, such as using a Sentiment Analysis model trained in data of compliments and complaints, a more robust database, and a NER model with better performance.

The second experiment, regarding the ASAE intelligent form, is a performance evaluation using a dataset of complaints collected by the original Electronic Complaints Book. First, the distribution of datasets in economic activities, competence, and the possibility of submission was presented. Next, the details of the experiments carried out were presented. The results obtained demonstrate that it is possible to reduce the entry of unwanted contacts. However, this may mean that, on the other hand, there is a loss of information relevant to the institution. Therefore, it is necessary to use classification models with better performance and new experiments to be carried out.

The analysis conducted in this chapter provides valuable information that can guide the development of the IF concept and validate some decisions made throughout the development phase.

# Chapter 6

# Conclusions



## 6.1 Summary

Many public and private institutions receive numerous contacts through online forms. The purpose of these forms is to facilitate communication between the user and the institution. Nonetheless, these typically have structured and open fields, leading to contacts that do not contain enough information or quality. In addition, the submissions obtained are processed manually or semi-automatic within the institution, which may require several inputs related to the contacted institution's activity domain.

Institutions are contacted for different reasons, whether to request information, make a complaint, or even acknowledge a well-provided service. Regardless, an individual that fills in a contact form is faced with several problems that can lead to low contact form usage. Therefore, the usability of the forms seeks to build methods that increase the communication, that is, that it produces, within the organization, the result sought by the complainant.

Furthermore, the user interface is a critical component of Human-Computer Interaction and is essential for user-organization interaction. Nevertheless, the form itself is regularly not comprehended by users, attending to defects in using the form and interacting with the organization. This contributes to the need for further client/user interactions with the purpose of collect supplementary information. When this does not happen, the contact is filed when it is unmanageable to follow up.

The solution to this problem involves creating the concept of Intelligent Forms that empower the user with tools that increase communication efficiency with the institution. Sending a contact should only be allowed if the information provided contains all the data necessary for the institution's good process.

This concept of Intelligent Forms needs to contain minimum requirements to ensure an improvement in communication, which is fundamental for solving the problem presented. The main conditions include validating information between fields, assistive writing tools, and machine learning models that allow textual content classification.

From the developed concept and the collected requirements, creating a tool that allows building Intelligent Forms was possible. This tool uses a drag & drop system and provides the user with a set of ready-to-use classification models.

Two forms were then developed from this same tool that allows the demonstration of the concept of Intelligent Form in practice. The first form was built within the scope of the Electronic Complaints Book to provide a solution to help fill in structured fields. In contrast, the second form was designed within the area of ASAE in order to assist the internal processing by the institution by previously classifying the content of the contact and verifying that it is following the domain of the institution's activity.

Experiments were then conducted in both case studies to verify the practical examples' integrity. The experiments carried out were different for each case, with the Electronic Complaints Book form being tested using volunteers who performed a task and assessed any problems and failures. In the ASAE form, a performance evaluation was carried out, using a dataset of complaints collected from the original Electronic Complaints Book.

Once the experiments were conducted, the results showed that, in general, the limitation on the effectiveness of communication between the user and the entity in question was slightly overcome. However, some improvements have to be applied to achieve better results. are obtained. In the experiment in which the user experience was tested, it was possible to conclude that classifiers with better performance are necessary to apply so that it is possible to overcome this limitation even further.

Regarding the experiments carried out on the ASAE form, it was possible to conclude that using two classification models and blocking the form submission as the result of these classifications helps prevent unwanted contacts from entering the institution. Notwithstanding, it is necessary to balance the use of these classifiers so that there is no loss of important information.

Finally, it is impossible to conclude with certainty that the experiments achieved their objective of determining whether the restriction on communication effectiveness between the user and the institution was overcome. However, it is possible to say that there was an improvement in this communication.

## 6.2 Challenges

The challenges of this research work began with the passage from an Intelligent Form concept to a practical example, and later to make possible a generalization of this concept to build a tool capable of developing several Intelligent Forms.

In the development of the tool, some challenges were also faced. The first was to find a mechanism or framework that allowed to build interfaces dynamically and that it was possible to adapt to implement the intended features. Furthermore, collecting machine learning models that allowed them to be easily applied was also challenging.

Concluding, the limitations were part of this work, shaping it and leading it towards different decisions than the initial proposal. However, the experiences were successfully executed and lead to exciting results for the problem stated in Chapter 1.

## 6.3 Future Work

This research study encountered several challenges, constraints, and procedural decisions, allowing concerns described in this section to be addressed in future work.

One of the main aspects for future work is applying assistive writing tools in the Intelligent Forms such as predictive typing, auto-completion, and grammar and spelling checkers to understand how they can improve the quality of the text that the user writes in unstructured fields. This application will then have to be evaluated to understand how it will help with human manual processing and the classification through machine learning classification models.

The machine learning models used were not trained with data related to the specific domain of the institutions in question. Therefore, it is relevant that, in subsequent phases, more specific models are used to improve the performance of these forms.

In addition to all this, new experiments can be done regarding usability and user experience to verify the functionalities of the developed forms and detect any problems. In the end, improvements can be made to the platform, providing a better user experience.

# Appendix A

# Distribution of complaints by Economic Activity and Competence

## A.1 Dataset distribution

| Economic Activity | # complaints | % |
|---|---|---|
| I - Primary production | 0 | 0 |
| II - Industry | 99 | 0.48 |
| III - Restaurant industry | 1549 | 7.50 |
| IV - Wholesalers | 58 | 0.28 |
| V - Retail | 3981 | 19.27 |
| VI - Direct selling | 0 | 0 |
| VII - Distance selling | 153 | 0.74 |
| VIII - Production and trade | 5755 | 27.91 |
| IX - Service providers | 4152 | 20.10 |
| X - Safety and environment | 3329 | 16.11 |
| Z - No activity identified | 1573 | 7.61 |
| Total | 20660 | 100 |

Table A.1: Economic Activity distribution

| Competence | # complaints | % |
|---|---|---|
| ASAE | 2683 | 12.99% |
| Other Entities | 17977 | 87.01% |
| Total | 20660 | 100 |

Table A.2: Competence distribution

# Appendix B

# Distribution of results by Economic Activity and Competence

## B.1 Distribution of evaluation results

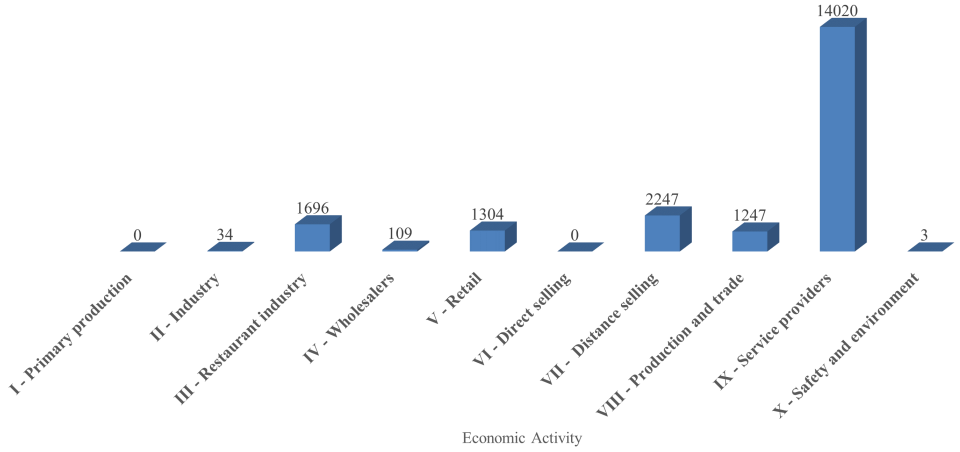


Figure B.1: Distribution of results by Economic Activity.

| | | Predicted values | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | I | II | III | IV | V | VI | VII | VIIi | IX | X |
| Actual Values | I | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | II | 0 | 0 | 14 | 0 | 1 | 0 | 1 | 0 | 24 | 0 |
| | III | 0 | 3 | 777 | 14 | 35 | 0 | 66 | 15 | 708 | 0 |
| | IV | 0 | 0 | 0 | 0 | 0 | 0 | 8 | 1 | 49 | 0 |
| | V | 0 | 18 | 387 | 25 | 1004 | 0 | 304 | 138 | 2105 | 0 |
| | VI | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | VII | 0 | 2 | 7 | 0 | 2 | 0 | 37 | 0 | 105 | 0 |
| | VIII | 0 | 5 | 197 | 35 | 92 | 0 | 772 | 626 | 4028 | 1 |
| | IX | 0 | 4 | 172 | 10 | 41 | 0 | 240 | 121 | 3563 | 1 |
| | X | 0 | 1 | 89 | 16 | 68 | 0 | 452 | 284 | 2418 | 1 |
| | Z | 0 | 1 | 53 | 9 | 61 | 0 | 367 | 62 | 1020 | 0 |

Table B.1: Confusion matrix of experiment with threshold 0.5 for Economic Activity task

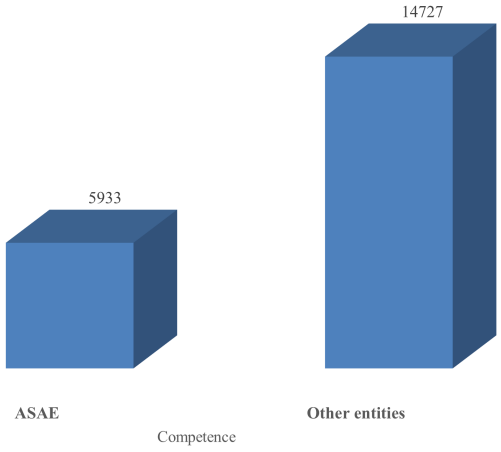


Figure B.2: Distribution of results by Competence.

| | | Predicted values | |
|---|---|---|---|
| | | ASAE | Other entities |
| Actual value | ASAE | 1106 | 1577 |
| | Other Entities | 4827 | 13150 |

Table B.2: Confusion matrix of experiment with threshold 0.5 for Competence task

# References


[1] S. Agarwal, A. Mohapatra, and M. R. Genesereth, "Smart forms," in *2016 AAAI Fall Symposia, Arlington, Virginia, USA, November 17-19, 2016*, AAAI Press, 2016.

[2] D. Sonntag, "Tutorial 3: Intelligent user interfaces," in *2015 IEEE International Symposium on Mixed and Augmented Reality, ISMAR 2015, Fukuoka, Japan, September 29 - Oct. 3, 2015* (C. Sandor, R. W. Lindeman, W. W. Mayol-Cuevas, N. Sakata, R. A. Newcombe, and V. Teichrieb, eds.), p. xxxii, IEEE Computer Society, 2015.

[3] P. P. Gaspar, "Direção da autoridade de segurança alimentar e económica." Available at https://www.asae.gov.pt/asae20/a-direcao.aspx, June 2019. (Accessed: January 12, 2021).

[4] L. Barbosa, J. Filgueiras, G. Rocha, H. L. Cardoso, L. P. Reis, J. P. Machado, A. C. Caldeira, and A. M. Oliveira, "Automatic identification of economic activities in complaints," in *Statistical Language and Speech Processing - 7th International Conference, SLSP 2019, Ljubljana, Slovenia, October 14-16, 2019, Proceedings*, pp. 249–260, 2019.

[5] D. Jurafsky and J. H. Martin, *Speech and language processing: an introduction to natural language processing, computational linguistics, and speech recognition, 2nd Edition*. Prentice Hall series in artificial intelligence, Prentice Hall, Pearson Education International, 2009.

[6] M. Eiband, S. T. Völkel, D. Buschek, S. Cook, and H. Hussmann, "When people and algorithms meet: user-reported problems in intelligent everyday applications," in *Proceedings of the 24th International Conference on Intelligent User Interfaces, IUI 2019, Marina del Ray, CA, USA, March 17-20, 2019* (W. Fu, S. Pan, O. Brdiczka, P. Chau, and G. Calvary, eds.), pp. 96–106, Acm, 2019.

[7] V. Alvarez-Cortes, V. H. Zarate, J. A. R. Uresti, and B. E. Zayas, "Current challenges and applications for adaptive user interfaces," in *Advances in Human Computer Interaction* (S. Pinder, ed.), ch. 2, Rijeka: IntechOpen, 2008.

[8] M. T. Maybury, "Intelligent user interfaces: An introduction," in *Proceedings of the 4th International Conference on Intelligent User Interfaces, IUI 1999, Los Angeles, CA, USA, January 5-8, 1999* (M. T. Maybury, P. A. Szekely, and C. G. Thomas, eds.), pp. 3–4, Acm, 1999.

[9] S. Shaikh, M. Sawand, N. A. Khan, and F. B. Solangi, "Comprehensive understanding of intelligent user interfaces," *International Journal of Advanced Computer Science and Applications*, vol. 8, 2017.

[10] K. Yoshida and H. Motoda, "Automated user modeling for intelligent interface," *Int. J. Hum. Comput. Interact.*, vol. 8, no. 3, pp. 237–258, 1996.

[11] B. A. Goodman and D. J. Litman, "On the interaction between plan recognition and intelligent interfaces," *User Model. User Adapt. Interact.*, vol. 2, no. 1-2, pp. 83–115, 1992.

[12] G. G. Hendrix, "Natural-language interface," *Am. J. Comput. Linguistics*, vol. 8, no. 2, pp. 56–61, 1982.

[13] R. Sharma, V. I. Pavlovic, and T. S. Huang, "Toward multimodal human-computer interface," *Proceedings of the IEEE*, vol. 86, no. 5, pp. 853–869, 1998.

[14] B. D. Davison and H. Hirsh, "Experiments in UNIX command prediction," in *Proceedings of the Fourteenth National Conference on Artificial Intelligence and Ninth Innovative Applications of Artificial Intelligence Conference, AAAI 97, IAAI 97, July 27-31, 1997, Providence, Rhode Island, USA*, p. 827, 1997.

[15] P. R. Cohen, M. Dalrymple, D. B. Moran, F. C. N. Pereira, J. W. Sullivan, R. A. G. Jr., J. L. Schlossberg, and S. W. Tyler, "Synergistic use of direct manipulation and natural language," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI 1989, Austin, Texas, USA, April 30 - June 4, 1989*, pp. 227–233, 1989.

[16] K. Tsiakas, M. Abujelala, A. Rajavenkatanarayanan, and F. Makedon, "User skill assessment using informative interfaces for personalized robot-assisted training," in *Learning and Collaboration Technologies. Learning and Teaching - 5th International Conference, LCT 2018, Held as Part of HCI International 2018, Las Vegas, NV, USA, July 15-20, 2018, Proceedings, Part II*, pp. 88–98, 2018.

[17] C. Stephanidis, C. Karagiannidis, and A. Koumpis, "Decision making in intelligent user interfaces," in *Proceedings of the 2nd International Conference on Intelligent User Interfaces, IUI 1997, Orlando, Florida, USA, January 6-9, 1997*, pp. 195–202, 1997.

[18] A. Bunt, C. Conati, and J. McGrenere, "Supporting interface customization using a mixed-initiative approach," in *Proceedings of the 12th International Conference on Intelligent User Interfaces, IUI 2007, Honolulu, Hawaii, USA, January 28-31, 2007*, pp. 92–101, 2007.

[19] E. Horvitz, "Principles of mixed-initiative user interfaces," in *Proceeding of the CHI '99 Conference on Human Factors in Computing Systems: The CHI is the Limit, Pittsburgh, PA, USA, May 15-20, 1999*, pp. 159–166, 1999.

[20] I. Rocketgenius, "Gravity forms features." Available at https://www.gravityforms.com/features/, Oct. 2019. Accessed: 2021-02-10.

[21] M. Olsha-Yehiav, M. B. Palchuk, F. Y. Chang, D. P. Taylor, J. L. Schnipper, J. A. Linder, Q. Li, and B. Middleton, "Smart forms: Building condition-specific documentation and decision support tools for ambulatory EHR," in *AMIA 2005, American Medical Informatics Association Annual Symposium, Washington, DC, USA, October 22-26, 2005*, AMIA, 2005.

[22] E. D. Liddy, "Natural language processing.," *Encyclopedia of Library and Information Science, 2nd Ed. NY. Marcel Decker, Inc.*, 2001.

[23] G. G. Chowdhury, "Natural language processing," *Annu. Rev. Inf. Sci. Technol.*, vol. 37, no. 1, pp. 51–89, 2003.

[24] A. Kadhim, "An evaluation of preprocessing techniques for text classification," *International Journal of Computer Science and Information Security,*, vol. 16, pp. 22–32, 06 2018.

[25] J. Camacho-Collados and M. T. Pilehvar, "On the role of text preprocessing in neural network architectures: An evaluation study on text categorization and sentiment analysis," in *Proceedings of the Workshop: Analyzing and Interpreting Neural Networks for NLP, BlackboxNLP@EMNLP 2018, Brussels, Belgium, November 1, 2018*, pp. 40–46, 2018.

[26] M. Toman, R. Tesar, and K. Jezek, "Influence of word normalization on text classification," *Proceedings of InSciT*, 01 2006.

[27] K. Kowsari, K. J. Meimandi, M. Heidarysafa, S. Mendu, L. E. Barnes, and D. E. Brown, "Text classification algorithms: A survey," *Inf.*, vol. 10, no. 4, p. 150, 2019.

[28] D. Yan, K. Li, S. Gu, and L. Yang, "Network-based bag-of-words model for text classification," *IEEE Access*, vol. 8, pp. 82641–82652, 2020.

[29] M. X. Chen, B. N. Lee, G. Bansal, Y. Cao, S. Zhang, J. Lu, J. Tsay, Y. Wang, A. M. Dai, Z. Chen, T. Sohn, and Y. Wu, "Gmail smart compose: Real-time assisted writing," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019, Anchorage, AK, USA, August 4-8, 2019*, pp. 2287–2295, 2019.

[30] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," *CoRR*, vol. abs/1810.04805, 2018.

[31] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving language understanding by generative pre-training," *OpenAI blog*, 2018.

[32] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners," *OpenAI blog*, vol. 1, no. 8, p. 9, 2019.

[33] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, "Language models are few-shot learners," in *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual* (H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, eds.), 2020.

[34] K. C. Arnold, K. Chauncey, and K. Z. Gajos, "Predictive text encourages predictable writing," in *IUI '20: 25th International Conference on Intelligent User Interfaces, Cagliari, Italy, March 17-20, 2020*, pp. 128–138, 2020.

[35] A. Fahda and A. Purwarianti, "A statistical and rule-based spelling and grammar checker for indonesian text," in *2017 International Conference on Data and Software Engineering (ICoDSE)*, pp. 1–6, 2017.

[36] J. J. Darragh, I. H. Witten, and M. L. James, "The reactive keyboard: A predicive typing aid," *Computer*, vol. 23, no. 11, pp. 41–49, 1990.

[37] A. Fowler, K. Partridge, C. Chelba, X. Bi, T. Ouyang, and S. Zhai, "Effects of language modeling and its personalization on touchscreen typing performance," in *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems, CHI 2015, Seoul, Republic of Korea, April 18-23, 2015*, pp. 649–658, 2015.

[38] K. Vertanen and P. O. Kristensson, "Intelligently aiding human-guided correction of speech recognition," in *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2010, Atlanta, Georgia, USA, July 11-15, 2010*, 2010.

[39] K. Vertanen, H. Memmi, J. Emge, S. Reyal, and P. O. Kristensson, "Velocitap: Investigating fast mobile text entry using sentence-based decoding of touchscreen keyboard input," in *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems, CHI 2015, Seoul, Republic of Korea, April 18-23, 2015*, pp. 659–668, 2015.

[40] S. Green, J. Chuang, J. Heer, and C. D. Manning, "Predictive translation memory: a mixed-initiative system for human language translation," in *The 27th Annual ACM Symposium on User Interface Software and Technology, UIST '14, Honolulu, HI, USA, October 5-8, 2014*, pp. 177–187, 2014.

[41] A. S. Arif, S. Kim, W. Stuerzlinger, G. Lee, and A. Mazalek, "Evaluation of a smart-restorable backspace technique to facilitate text entry error correction," in *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems, San Jose, CA, USA, May 7-12, 2016*, pp. 5151–5162, 2016.

[42] A. Kannan, K. Kurach, S. Ravi, T. Kaufmann, A. Tomkins, B. Miklos, G. Corrado, L. Lukács, M. Ganea, P. Young, and V. Ramavajjala, "Smart reply: Automated response suggestion for email," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, pp. 955–964, 2016.

[43] D. Ward, J. Hahn, and K. Feist, "Autocomplete as research tool: A study on providing search suggestions," *Information Technology and Libraries*, vol. 31, 12 2012.

[44] A. Nandi and H. Jagadish, "Assisted querying using instant-response interfaces," in *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pp. 1156–1158, 01 2007.

[45] R. White and G. Marchionini, "Examining the effectiveness of real-time query expansion," *Inf. Process. Manage.*, vol. 43, pp. 685–704, 05 2007.

[46] X. Tong and D. A. Evans, "A statistical approach to automatic OCR error correction in context," in *Fourth Workshop on Very Large Corpora, VLC@COLING 1996, Copenhagen, Denmark, August 4, 1996*, 1996.

[47] R. Amorim, *An Adaptive Spell Checker Based on PS3M: Improving the Clusters of Replacement Words*, vol. 57, pp. 519–526. Springer, Berlin, Heidelberg, 12 2009.

[48] A. J. Carlson, J. Rosen, and D. Roth, "Scaling up context-sensitive text correction," in *Proceedings of the Thirteenth Innovative Applications of Artificial Intelligence Conference, August 7-9, 2001, Seattle, Washington, USA*, pp. 45–50, 2001.

[49] N. Bhirud, R. Bhavsar, and B. Pawar, "Grammar checkers for natural languages : A review," *International Journal on Natural Language Computing*, vol. 6, pp. 1–13, 08 2017.

[50] D. Naber, *A Rule-Based Style and Grammar Checker*. Citeseer, 08 2003.

[51] B. Manchanda, V. A. Athavale, and S. kumar Sharma, "Various techniques used for grammar checking," *International Journal of Computer Applications & Information Technology*, vol. 9, no. 1, p. 177, 2016.

[52] M. Dorash, "Machine learning vs. rule based systems in nlp." Available at https://medium.com/friendly-data/machine-learning-vs-rule-based-systems-in-nlp-5476de53c3b8, Dec. 2017. (Accessed: January 20, 2021).

[53] M. Soni and J. S. Thakur, "A systematic review of automated grammar checking in english language," *CoRR*, vol. abs/1804.00540, 2018.

[54] J. Peng, K. Lee, and G. Ingersoll, "An introduction to logistic regression analysis and reporting," *Journal of Educational Research - J EDUC RES*, vol. 96, pp. 3–14, 09 2002.

[55] H. Sharma and S. Kumar, "A survey on decision tree algorithms of classification in data mining," *International Journal of Science and Research (IJSR)*, vol. 5, 04 2016.

[56] B. Gupta, A. Rawat, A. Jain, A. Arora, and N. Dhami, "Analysis of various decision tree algorithms for classification in data mining," *International Journal of Computer Applications*, vol. 163, pp. 15–19, 04 2017.

[57] S. Chakrabarti, S. Roy, and M. V. Soundalgekar, "Fast and accurate text classification via multiple linear discriminant projections pp," *VLDB J.*, vol. 12, no. 2, pp. 170–185, 2003.

[58] I. Rish *et al.*, "An empirical study of the naive bayes classifier," in *IJCAI 2001 workshop on empirical methods in artificial intelligence*, vol. 3, pp. 41–46, 2001.

[59] T. Joachims, "Text categorization with support vector machines: Learning with many relevant features," in *Machine Learning: ECML-98, 10th European Conference on Machine Learning, Chemnitz, Germany, April 21-23, 1998, Proceedings*, pp. 137–142, 1998.

[60] A. Bilski, "A review of artificial intelligence algorithms in document classification," *International Journal of Electronics and Telecommunications*, vol. 57, pp. 263–270, 09 2011.

[61] D. Pisner and D. Schnyer, *Support vector machine*, pp. 101–121. Elsevier, 01 2020.

[62] J. Dukart and F. Roche, "Basic concepts of image classification algorithms applied to study neurodegenerative diseases," *Brain Mapping: An Encyclopedic Reference*, vol. 3, pp. 641–646, 12 2015.

[63] H. Bhavsar and M. H. Panchal, "A review on support vector machine for data classification," *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)*, vol. 1, no. 10, pp. 185–189, 2012.

[64] H. Chung, S. J. Lee, and J. G. Park, "Deep neural network using trainable activation functions," in *2016 International Joint Conference on Neural Networks, IJCNN 2016, Vancouver, BC, Canada, July 24-29, 2016*, pp. 348–352, 2016.

[65] I. Sutskever, J. Martens, and G. E. Hinton, "Generating text with recurrent neural networks," in *Proceedings of the 28th International Conference on Machine Learning, ICML 2011, Bellevue, Washington, USA, June 28 - July 2, 2011*, pp. 1017–1024, 2011.

[66] R. Johnson and T. Zhang, "Effective use of word order for text categorization with convolutional neural networks," in *NAACL HLT 2015, The 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Denver, Colorado, USA, May 31 - June 5, 2015*, pp. 103–112, 2015.

[67] K. Chatsiou, "Text classification of COVID-19 press briefings using BERT and convolutional neural networks," *CoRR*, vol. abs/2010.10267, 2020.