UNIVERSITÀ DEGLI STUDI DI PADOVA

Dipartimento di Fisica e Astronomia "Galileo Galilei"

Master Degree in Physics of Data

Final Dissertation

Improving Data Quality in Customer Relationship Management Systems: a method for cleaning personal information

Internal Supervisor

Alberto Garfagnini

External Supervisor

Daniele Miorandi

Candidate

Elena Leonelli

Academic Year 2022/2023

Abstract

In recent years, data have become crucial in areas such as marketing, sales and, more generally, in companies. Customer Relationship Management (CRM) systems are used by companies to manage the interactions with potential and existing customers, with the aim of improving their experience and increasing sales. Such systems are based on the collection and processing of large amounts of data, which are used to build tailored customer solutions. However, the effectiveness of this approach is also determined by the quality of the data available, and its increase has a direct impact on overall business performance. In this thesis, a method is developed to improve the quality of personal customer data from a CRM system. The method is designed to clean datasets with attributes such as name, surname and email address, and exploits machine learning algorithms, natural language processing techniques, and rule-based approaches to improve Data Quality. To evaluate its effectiveness, the proposed method has been tested on a synthetic dataset specifically created, showing a significant improvement in Data Quality. Moreover, its performance was also compared to that of an existing method, and results show that the method developed in this thesis is more effective.

Glossary

attribute the column of a dataset. Attributes are Name (First Name), Surname (Last Name), Email (Email address).

clean dataset the version of the dataset after the action of a cleaning method.

contact the row of a dataset; a contact can have multiple attributes. It is a synonym of record.

correct referring to the content of a cell, it is correct when it coincides with the content of the same cell in the ground truth dataset.

correctly cleaned the content of a cell is correctly cleaned when before the action of the cleaning method it is not correct, but after it results correct.

datumo SaaS developed by U-Hopper Srl, performing two tasks: clean and enrich. If not specified, it is understood to refer only to the cleaning task, and it is a cleaning method for contacts in a CRM system dataset.

dirty dataset the version of the dataset not 100% correct; the version on which a cleaning method acts on.

entity is the largest string in a name or surname containing no spaces; one or more entities form a name or surname (e.g. the name "Anna Maria" is formed by two entities: "Anna" and "Maria").

fill is one action that the cleaning method can perform, consisting in adding some content to a previously empty cell.

general email the type of email address when it is not specifically referring to a person but rather to an office (e.g. *sales@u-hopper.com*).

ground truth dataset the 100% correct version of the dataset; this version is not provided for all datasets.

internal dataset the sum of "Daniele Hubspot" dataset and "Thinkin" Dataset. This dataset is completely known (no GDPR issues) and has both the dirty and the ground truth version. It can be subdivided in two subsets: "internal nominative" and "internal general", according to the contacts email type.

leave unchanged is one action that a cleaning method can perform, consisting in not changing the content of a cell (both in the case of an empty cell or a non-empty cell). It is a synonym of "no action".

modify is one action that a cleaning method can perform, consisting in changing the content of a previously non-empty cell.

name the content of a cell with attribute Name; it is formed by one or more entities.

Datatect cleaning method for contacts in a CRM system dataset developed in this thesis.

nominative email the type of email address when it is referring to a person (e.g. *giovannirossi@u-hopper.com*).

record the row of a dataset; a record can have multiple attributes. It is a synonym of contact.

size the number of rows in a dataset, which corresponds to the number of contacts.

surname the content of a cell with attribute Surname; it is formed by one or more entities.

synthetic dataset the dataset "created in the lab" used for testing a cleaning method.

users dataset the sum of the six datasets loaded into datumo. This dataset is subject to GDPR, therefore and it is not fully provided, but only with some previously collected statistics based on its dirty version and on its clean version (after datumo's action).

Contents

Chapter 1

Introduction

The growing importance of data in human activities is undeniable. The data generated, collected, and stored have grown exponentially in recent years, thanks to the widespread adoption of digital technologies and the internet, as well as the development of data storage and processing technologies. From the way we communicate to the way we make decisions, data play a crucial role in shaping our understanding of the world around us.

One of the most significant areas where data are playing an increasingly vital role is in business and marketing. Data-driven decision-making has become fundamental for companies of all sizes, and businesses are leveraging data to gain a competitive advantage in the marketplace. By analyzing customer data, businesses can gain insights into consumer behavior and preferences, which can be used to develop targeted marketing campaigns and improve customer loyalty.

One of the methods used by data-driven companies is Customer Relationship Management (CRM). Born in the 1990s, Customer Relationship Management is a business strategy that aims to improve enterprise performance through a customer-focused approach. The relationship with current and potential customers is created and maintained through the collection of customer data, which help in interpreting customer needs in order, for example, to perform better marketing campaigns.

Examples of Customer Relationship Management systems are software that can integrate social media and mailboxes, providing insights and analytics based on data collected, offer marketing campaigns automation, and more. The goal is to help companies grow, by developing a data-driven approach that combines customer services, marketing and sales departments, and content management in one place.

CRM systems are based on different types of data: users personal information, such as name, address, email address; users behaviour information, for example purchase history and buying preferences; data concerning company turnover and so on. The efficacy of CRM systems depend on the quality of those data [1].

As a solution for data quality issues, U-Hopper Srl developed datumo, a SaaS (Software as a Service) that aims to improve the quality of users personal data. On each contact of the CRM system dataset, datumo performs two actions: on the one hand it cleans personal information (name and surname), on the other hand it enriches them by adding attributes such as gender, language, company. The cleaning task is performed by correcting possible errors in Name and Surname attributes, also using information that can be retrieved from email address (which is assumed to be correct).

This thesis is the result of an internship at U-Hopper, during which a cleaning method for the contacts of a CRM system dataset was developed. The method was designed to be a development and improvement on the one already used by datumo, with the aim of helping to further increase the data quality of the CRM system. The method developed, Datatect, exploits various techniques and approaches to clean personal data.

In the second chapter of this thesis, the definition of data quality, the theoretical context on which Datatect is based, and the data underlying its development are presented. The third and the fourth chapter are devoted, respectively, to the performance evaluation methodology and the construction of a synthetic dataset used for the evaluation. The fifth chapter contains the description of the structure of Datatect, while in the sixth chapter the performance results are reported. The last chapter is dedicated to conclusions and directions for future developments.

Chapter 2

Background

The cleaning of a CRM system, *i.e.* the process by which its overall Data Quality is increased, consists of several tasks. No evidence of a similar problem could be found in the literature, given the specificity of the data present (personal names and email addresses) and of the way the problem is posed. However, inspiration for the development of specific tasks such as language identification and word segmentation was taken from various research, which contributed to the design of the proposed solution.

This chapter presents the ideas and techniques behind the development of Datatect, as well as a brief introduction to the concept of Data Quality.

2.1 Data Quality

The wide variety of data types, as well as the many different contexts in which these are used, makes it hard to find a universal definition of Data Quality (DQ). Among all the different characterizations present in literature, one of the most common is the one that relates the term "quality" with the concept of "fitness for use": the more a data collection satisfies its consumer's requirements, the higher the quality [2]. This framework emphasizes the central role of the user: the degree of quality is strictly connected to the context and the purposes requiring those data.

According to the SQuaRE series of International Standards (*Systems and software Quality Requirements and Evaluation*), the definition of data quality is the following [3]:

> *Data Quality is the degree to which the characteristics of data satisfy stated and implied needs when used under specified conditions.*

Data quality has a multidimensional nature: there may be different issues affecting a data set and causing low data quality. As an example, consider a table with data of a graduation session (see *Table* 2.1). Each row is a record and corresponds to a single participant; the columns are the attributes that describe candidate's name, supervisor, thesis title and graduation date.

In the table, there are typos ("Annna" instead of "Anna", "12.04.2003" intead of "12.04.2023"), mix-ups between attributes of the same record (in the first row, "Prof. Giorgio Anesi" and "Annna Rossi"), and missing values (full or partial, as in the case of "Francesca").
Moreover, by looking attribute-wise one can find other inconsistencies: the Candidate full name is written both as name-surname ("Emma White") and as surname-name ("Bianchi Guglielmo"); the

Candidate	Supervisor	Thesis Title	Date
Prof. Giorgio Anesi	Annna Rossi	xxxxxxxx	12.04.2023
Bianchi Guglielmo		Analysis of CRM data	April 13, 2023
Emma White	Prof. Emilia Dusini	13.04.2023	
Francesca	Prof. Giulia Gadda		12.04.2003

Table 2.1: Example of data quality issues in a data set.

Thesis Title should be not only a string of word characters ("13.04.2023" is clearly not a title), but also a meaningful one (unlike "xxxxxxxx"); finally, in the Date attribute there is no agreement on the format used ("12.04.2023" and "April 13, 2023").

Each of these errors contribute to worsen the data quality of the table in a different way; to have an overall improvement, one has to correctly identify and solve each of them. The multidimensionality consists in the coexistence of all these issue-types that, taken together, constitute the general concept of data quality. Each dimension represents a type of issue, and has its own characteristics which determine how to identify the problem, and the possible ways to solve it.

2.1.1 Data quality dimensions

To capture all data quality facets a number of attributes that represent a single aspect of data quality, called data quality dimensions, has been defined. As for the definition of data quality, there is no agreement in literature on its dimensions: different researchers have different opinions on what can be considered an appropriate set of data quality dimensions.

In the 1990s, Wang and Strong [2] developed a hierarchical framework for organizing data quality dimensions, finding 20 dimensions grouped into four categories: intrinsic, accessibility, contextual, and representational. Redman [4] proposed a 15-dimensional set, within the same four categories. In the context of data warehouses, Jarke [5] identified completeness, credibility, accuracy, consistency and data interpretability as properties directly autoreferring to the stored data. The data quality framework proposed by Bovee *et al* [6] consists of four essential attributes (accessibility, interpretability, relevance, integrity) and four subattributes of integrity (accuracy, completeness, consistency, existence). Integrity, and its subattributes, are properties intrinsic in the nature of data, while accessibility, interpretability, relevance are extrinsic.
Among all these different set of attributes for data quality, it is possible to access that accuracy, completeness, consistency, timeliness, interpretability and accessibility are the six most cited dimensions.

This thesis considers the data quality dimensions as defined in the "Data quality model" (ISO/IEC 25012) developed *for data retained in a structured format within a computer system* [7]. This model is part of the *Quality Model Division* of the SQuaRE (*Systems and software Quality Requirements and Evaluation*) series of International Standards. According to this ISO standard, data quality has 15 main characteristics, divided into inherent and system-dependent. Inherent characteristics are those that describe the intrinsic potential of the data to satisfy stated and implied needs. On the other side, system dependent characteristics are related to the capability of the computer system to reach and preserve data quality. This classification allows to separately identify the two different aspects of data quality: one dependent on data itself, and the other related to the technologies used to store and access data.

The fifteen dimensions of data quality are:

Inherent Accuracy, Completeness, Consistency, Credibility, Currentness;

Inherent and system-dependent Accessibility, Compliance, Confidentiality, Efficiency, Precision, Traceability, Understandability;

System-dependent Availability, Portability, Recoverability.

The SQuaRE has also a *Quality Measurement Division*, and in particular the "Measurement of data quality" (ISO/IEC 25024), providing *measures including associated measurement methods and quality measure elements for the quality characteristics in the data quality model* [3]. In this framework are listed the data quality measures that can be used to monitor each of the 15 dimensions of the data quality model, with the purpose of defining the overall data quality of the system.

2.1.2 Data Quality dimensions in Datatect

This thesis focuses on the inherent data quality dimensions, since accessing data quality from a system-dependent point of view exceeds the scope of datumo (and consequently of Datatect). In particular, the data quality dimensions considered are accuracy, completeness and consistency: the remaining inherent characteristics are related to areas where Datatect does not act, such as the time updating (currentness) and the authenticity (credibility) of data.

In this section are reported the ISO definitions of the three data quality dimensions considered, together with the associated data quality measures that are significant in this context. The metrics used to assess data quality in this thesis were developed from these measures (see *Chapter* 3).

Accuracy

Accuracy is defined as *the degree to which data has attributes that correctly represent the true value of the intended attribute of a concept or event in a specific context of use* [7]. It concerns both the semantic and the syntactic sphere: the name "Marta" recorded as "Martq" has a low syntactic accuracy, while the same name recorded as "Anna" has a low semantic accuracy.

The measures of accuracy are:

- Syntactic data accuracy (Acc-I-1), evaluated as the ratio of syntactically accurate items over the total;
- Semantic data accuracy (Acc-I-2), which is the ratio of how accurate the data values are in terms of semantics.

Completeness

Completeness is *the degree to which subject data associated with an entity has values for all expected attributes and related entity instances in a specific context of use* [7]. An example of completeness issue is when a missing value is stored where a non-null one is expected.

The measures of completeness are:

- Record completeness (Com-I-1): the ratio of the number of non-empty items among all the attributes in a data record over the total;

- Attribute completeness (Com-I-2): the ratio of non-empty items related to the same attribute, over the total items for that attribute;
- Empty records in a data file (Com-I-5): the ratio of the number of empty records over the total number of records in a data file.

Consistency

Consistency is defined as *the degree to which data has attributes that are free from contradiction and are coherent with other data in a specific context of use* [7]. This dimension is meant to be either or both within data of the same record, and across attributes. In the former case, there is low consistency for example when one record has "Green" as attribute Surname, and "John Snow" as Full Name; in the latter case, a consistency issue occurs when the attribute Birth Date is "02/01/2000" for one record and "January 2, 2000" for another.

Consistency can be measured with:

- Semantic consistency (Con-I-6): the degree to which semantic rules are respected.

2.2 Language identification

Automatic Language Identification (LI) is the task of natural language processing (NLP) that aims to identify the language of a given text or speech [8]. In principle, automatic language identification applies to any type of text and modality of language: written documents (hand-written or digital), audio files, signs language. However, for the purposes of this thesis the following discussion is limited to language identification applied to written texts stored in a digital form.

LI models seek to mimic the human ability to recognize languages or at least the language group they belong to, even without being able to speak the language or without knowing many of its words. In fact, not only a bilingual person is able to recognize whether a document is written in one or in the other language, but this ability extends also to people not having a very thorough knowledge of the language. Most Europeans will be able to state that the phrase *Ich stehe frühmorgens auf, um zu joggen und gehe dann ins Büro* is written in German: the presence of words such as *ich* and *und*, and the presence of umlaut, are in principle sufficient to identify this language.

It is nevertheless necessary to have some familiarity with a language in order to be able to correctly identify it: a Chinese person who knows neither German nor Dutch would probably not be able to distinguish between them, just as for an European the difference between Chinese and Japanese is very limited. Automatic LI can exceed this human limit, being theoretically capable of recognising all human languages. However, it is also true that the more similar two languages (or even dialects) are, the more difficult automatic LI is to perform.

2.2.1 Features for language identification

There are many mechanisms underlying the human ability of language identification: as shown in the previous example, two of them can be the detection of "revealing words" or the presence of particular diacritical marks. Depending on the application, LI models use many strategies in order to perform the identification; the principal categories of features used in literature are listed in the following [8].

Single-character tokens

One of the set of features used in LI is single-characters occurrences, based on the idea that the frequency of a single character in a document is language-specific. In 1974, Rau [9] was one of the first to use the probability of single characters occurrences as a feature to identify the language of a short sample text written in English or Spanish. Following his intuition, many researches include single-character tokens among the features considered in the LI model [10][11].

Other single-character approaches include the format of numbers as discriminating feature (for example between Malay and Indonesian [12]), and the alphabet-specific characters detection: back to 1989, Henrich [13] method for language identification in words made an initial skim thanks to the presence of language-specific characters in a word.

Multiple-character combinations

The division of letters into vowels and consonants has led to their relationship being regarded as a feature from the earliest studies in LI: Rau [9] studied the frequencies of vowels following vowels, vowels following consonants, consonants following vowels and consonants following consonants in English and Spanish.

However, the most used set of features falling under "multiple-characters combinations" type is character *n*-gram. A character *n*-gram is a contiguous sequence of n characters (either overlapping or consecutive) from a given string: the consecutive character bigrams of the word *hand* are *ha* and *nd*, while the overlapping ones are *ha*, *an* and *nd*. The overlapping *n*-grams are the most used in literature.

Beesley [14], Dunning [15] and Cavnar and Trenkle [16] were among the first to apply character *n*-grams to LI, in particular using statistical methods. More recently, *n*-grams have been used as features to train supervised machine learning models such as Support Vector Machines (SVMs) [17] [18], or Conditional Random Fields (CRFs) [19].

Single words

One strategy for language identification consists in comparing words in a document with an *ad hoc* built dictionary for each language. It has been shown that is not necessary to use a complete dictionary for each language, but instead create a subset containing the most revealing words, such as determiners, conjunctions and prepositions (because they are very common words) or unique words (words occurring in only one language). Some examples of this can be found in research from the 1990s: Wechsler [20] chose stop-words, Souter *et al* [21] build a dictionary of the most common words, Grefenstette [22] listed the most frequent short words (5 or less characters) per language.

Multiple-words combinations

Languages differ not only in terms of vocabulary, but also in terms of syntax and parts of speech: from this insight follows the use of groups of words as features. Word *n*-grams are often used in LI, sometimes together with character *n*-grams as in [23]. Investigating syntactic constructs is useful when two dialects or similar languages have to be distinguished: Laboreiro *et al* [24] found that the difference in verbal forms between Brazilian Portuguese and European Portuguese helps in their distinction.

2.2.2 Language identification for personal names

Despite being in some cases considered "a solved task" [25], language identification is not always straightforward. In the simplest case of a monolingual long-enough document, many of the features presented above may seem quite redundant: a dictionary of the 100 most-frequent words for each language may suffice. However, in the last two decades LI has been applied to a variety of contexts, each requiring a specific combination of features.

This thesis requires a sub-task of LI that deals with short texts and personal names: language detection based on full names. The intuition behind is that personal names have language-dependent characteristics: as humans, we can quite easily access that "Marco Rossi" is more likely to be an Italian name, whereas "Heidi Schmidt" is likely a German one[1]. Personal names have sufficient clues to identify the language they come from, even more than general words [26]: with the right choice of model and feature set, good results can be achieved for this task.

The challenge of language identification for personal names is made difficult firstly by the length of the input. As shown by Baldwin and Lui [27], the task of LI becomes increasingly difficult the shorter the length of the document. Moreover, for this reason many of the features presented above can not be used, or are not suitable for this application. Most of the research in this field uses overlapping *n*-grams at character level to represent the full-names: this choice guarantees a sufficient number of features as input to the model, although the full-names length of around 15 characters.

In second place, names are very special words, and they have inherent properties that may compromise LI: it is not uncommon for surnames and first names to belong to two different linguistic spheres, or for them to be so peculiar that they do not resemble any language. These intrinsic features can not be easily eliminated and must be taken into account, because they can introduce noise into the model.

Furthermore, personal names are a separate category from the general words of a language: it has been demonstrated that LI models for personal names achieve better performances when trained on names, rather than on a general set of words [26]. This precludes the possibility of using the numerous models and libraries for LI already present, as well as general data and dictionaries for each language [28] [29].

LI for personal names has been investigated in relation to word pronunciation: personal names are among the most mispronounced words, and including their ethnic origin has been shown to improve pronunciation results [30] [31]. Li *et al* [32] use LI in a transliteration system to take into account the different semantic transliteration rules between languages.

Nobesawa and Tahara [33] developed a LI statistical model based on character unigrams, bigrams, trigrams and length of the full names; the system evaluates the probability for an input name to belong to a language using only statistical data extracted from the names corpus (130840 full names for 9 languages). This model reached over 90% of accuracy for Japanese, Korean and Russian full names, showing not only that personal names have language-specific *n*-grams frequencies, but also that the length of personal names (16.19 on average) is enough to allow language identification.

Bhargava and Kondrak [17] proposed to use character *n*-gram counts and length as features to train a Support Vector Machine (SVM). This approach has been tested on two different datasets: the Transfermarkt corpus [26], which consists of 14915 European soccer player full names (14.8 average

[1]The language identified is not meant to be the language that the owner of the name speaks, nor his nationality, but rather the language his name origins from.

name length) and 13 possible national languages; and the Chinese-English-Japanese corpus [32] with 97115 among names and surnames (7.6 average name length). The first dataset reached 79.9% of overall accuracy with Linear SVM and *n*-grams with n up to 5; with Linear SVM and n up to 4, the second dataset reached 97.6% of overall accuracy. This work suggests that language identification for personal names can benefit from a machine learning approach that exploits language-specific *n*-grams frequencies.

In this thesis work, a language identification model for personal names was developed from the insights offered by the research cited above. As will be explained in more detail in *Chapter* 5, the approach adopted uses character bigrams and trigrams as input features for a SVM.

2.3 Word segmentation

In natural language processing, word segmentation (WS) is the task of inserting word boundary characters in order to subdivide an input string into its component words. WS is a challenging task especially for languages having no explicit word delimiters, such as Chinese, Thai, Japanese: for this languages, any NLP task must firstly address the problem of word segmentation. Research in Chinese word segmentation (CWS) in particular is very active [34], and most of the efforts in the field of word segmentation are devoted to it. CWS (and generally word segmentation) algorithms can be divided into three main categories: methods based on dictionaries, methods based on understanding, methods based on statistics [35] [36].

Dictionaries or string-matching methods recognise words within the text by comparing them with a sufficiently large machine dictionary, and according to a certain strategy. The text is scanned either forward, backward, or forward and backward, and a string is recognised as a word whenever it successfully matches with a term in the dictionary.

Word segmentation based on semantic and syntactic understanding exploits deep learning techniques, simulating the process of human understanding of the sentence and using this information to improve word segmentation. In the last years several methods to model the contextual information were investigated, such as Recursive Neural Networks, long-short term memory (LSTM) networks, and adversarial learning. Deep learning techniques usually require a large amount of linguistic knowledge and information, and thus a large amount of data.

Eventually, methods based on statistics rely on the idea that the more two words appear next to each other, the more likely they are to constitute a word. The reliability of a word is evaluated through the statistics of the frequency of the combination of adjacent co-occurrence words in the text. Statistics-based word segmentation methods include Hidden Markov models, Maximum entropy models, Conditional Random Fields (CRF).

2.3.1 Email address segmentation

In this thesis, word segmentation techniques are needed in order to extract single names and surnames when they are present in the local part of the email address[2], without any space or punctuation mark between them. This means extracting "john" and "green" as separate substrings from the string *johngreen*. This task is a very specific case of WS, and quite different with respect to Chinese word segmentation; therefore many techniques are not applicable. Firstly, the input in most word segmentation problems is a document, the length of which is much longer than the local part of an

[2]An email address is of the form {local_part}@{domain}.

e-mail address, making word segmentation methods based on understanding and statistics mostly unfeasible. In second place, Asian languages differ from those based on the Latin alphabet in terms of the length of each word, which is much shorter; methods that attempt to capture the dependency on previous characters (CRF, LSTM) are therefore more difficult to implement [37].

The approach used for local part segmentation is the one proposed by Norvig [38]. This is a dictionary-based probabilistic model that relies on a corpus containing words (in this case, names and surnames) with occurrence probabilities. These probabilities make possible the segmentation of a string allowing the calculation of probabilities for each candidate set of substrings; the best candidate c^* in the set of all possible candidates C is chosen as the one with the highest probability $P(c)$:

$$c^* = argmax_{c \in C} P(c).$$

The probabilities $P(c)$ are calculated from the product of the probability of each of the N substrings $P(s_k)$ forming the candidate c:

$$P(c) = \prod_{k=1}^{N} P(s_k).$$

As an example, consider the string *johngreen*. The candidate [*john*,*green*] has a higher probability than [*joh*, *ngr*, *een*] because the product of its components probabilities P(*john*)×P(*green*) is higher: of course the words *john* and *green* occur more often than *joh*, *ngr* or *een*.

It is unrealistic that the corpus contains all the possible names and surnames, therefore a management of the unknown words is needed. This is based on the distribution of names and surnames length, as will further explained in *Chapter* 5.

2.4 Personal names corpora

Datatect and its development are based on a large amount of data, especially personal names and surnames. In this section the names corpora available are reviewed, and the corpus used is presented.

Transfermarkt corpus It is based on the Transfermarkt website [39], that contains data on football competitions, clubs and players, with names and nationalities. Konstantopoulos [26] and later Bhargava and Kondrak [17] used a subset of it with 14915 football players from 13 countries. This data are still collected and automatically updated once a week [40], having 27919 players for different countries of origin (updated to 06/03/2023). An issue with this dataset is due to the fact that the players are only male, and that are not provided names and surnames separately, but rather full names. For these reasons, this corpus has not been used in this work of thesis.

NameDataset It is a python library developed by Phillipe Remy [41] containing 730 thousands first names and 983 thousands last names, extracted from 533 million Facebook users during a data leak of the social network. Starting from those data, the library provides statistics of names and surnames, including spread rates over 106 countries. This dataset is massive, but the fact that names are non-transliterated must be taken in account, as well as the noisiness of data. In fact, there is no procedure to ensure that the name given by a user is his or her real name, or that the nationality is as declared. As a consequence, the datasets of each country may contain many nicknames and invented names, and many names in languages not corresponding to the country, which introduce noise into the datasets themselves.

NameDataset has been used in this thesis for the creation of fake data, as will be explained in *Chapter* 4. In that context, the presence of noise in the dataset is exploited to obtain more realistic fake names and surnames.

2.4.1 Wikidata names corpus

The main data source used in this thesis is Wikidata [42], a free and multilingual database hosted by the Wikimedia Foundation that collects structured data. Names and surnames occurrences among people speaking 13 different languages were gathered, building 13 separate dictionaries: Arabic, Chinese, Dutch, English, French, German, Greek, Italian, Japanese, Portuguese (European and Brazilian), Russian, Spanish, Turkish. All the names were collected with English as wikibase language to avoid transliteration issues.

The dataset of each language is the result of a query to Wikidata, written in SPARQL language. The database was queried to provide names or surnames (property P735 *given name* and P734 *family name* respectively) of persons speaking a language (property P1412 *languages spoken, written or signed*). Full names are collected from item labels.

The choice of collecting data per language instead of per nationality is motivated by the fact that some countries have more than one official languages (Belgium, Canada..), and some languages are spread over multiple countries (German is spoken both in Germany and in Austria).

The dataset consists of 73635 unique names and 198117 unique surnames, preprocessed in order to remove diacritic marks (so that "José" is considered the same name as "Jose"). From the same data, a second dataset has been created with full names subdivided per language. The size of this second dataset has been limited to 30000 full names for each language.

A first assumption made in this thesis is that the noise of each language corpus is small enough to notice significant differences between them, and this is more realistic the larger the size of the dataset. A source of noise in this context is when the spoken language of a person does not match the etymological and linguistic origin of his name, thus introducing spurious names in the datasets (whether a name belongs to a dataset depends on the language spoken by its owner). A perhaps even stronger hypothesis is that these data reflect the actual distribution of names and surnames for each language. Although this is not true in an absolute sense, the two distributions are linked: the more Anna's there are, the more likely it is that some of them are present on Wikidata.

In *Table* 2.2 is shown the composition of the 13 languages corpora making up the Wikidata names corpus. Each language corpus is provided with references used to query Wikidata (item IDs, item labels), along with the sizes in terms of names, surnames, and full names. The size of some language corpora is small: Chinese and Greek have around two thousand unique entities between names and surnames, while English has over fifty thousand of entities. These differences are reflected in the different behaviour of the various languages in the models developed (see *Chapter* 5 for further details).

Language	Code	Item ID	Item label	Full names	Names	Surnames
Arabic	ar	Q13955	Arabic	27634	2587	2842
Chinese	zh	Q7850	Chinese	6309	1135	717
Dutch	nl	Q7411	Dutch	26744	3885	7868
English	en	Q1860	English	30000	21965	29589
French	fr	Q150	French	30000	9109	28284
German	de	Q188	German	30000	8986	48188
Greek	el	Q36510	Modern Greek	11254	1030	1404
Italian	it	Q652	Italian	30000	5753	34418
Japanese	ja	Q5287	Japanese	30000	2504	3804
Portuguese	pt	Q5146 + Q750553	Portuguese + Brazilian-Portuguese	30000	4106	6318
Russian	ru	Q7737	Russian	30000	3660	10834
Spanish	es	Q1321	Spanish	30000	6460	20123
Turkish	tr	Q256	Turkish	23643	2455	3728

Table 2.2: Data collected from Wikidata from people grouped by property P1412 *languages spoken, written or signed*: unique Full names, Names and Surnames. Note that Names and Surnames are the totality of the data present on Wikidata, while Full Names are in some cases a subset of them. Wikibase lang: en.

Chapter 3

Evaluation methodology

The purpose of Datatect is to increase the overall data quality of datasets from a CRM system; this occurs when the data quality of a dataset cleaned by Datatect is higher than the data quality of the original dataset. This chapter presents the metrics and methodologies that make this comparison possible.

After the first section where the problem is better defined, the first part of this chapter is dedicated to the metrics and methodologies used to evaluate the cleaning capacity of Datatect during its development. Thanks also to the definition of a key performance indicator (KPI), the impact of changes made from time to time on Datatect's ability to increase data quality was monitored.

In the last section are presented the metrics and methodologies used to evaluate the data quality of a user dataset uploaded to Datatect. As will be discussed in more detail below, this second case differs from the previous case since a 100% correct version of the dataset is not available for comparison.

3.1 Assessing Data Quality in a CRM system dataset

A dataset of a CRM system consists of of 3 columns (Name, Surname, Email), N rows and $3N$ cells. Each row contains data relating to a specific contact, whose email address can be *nominative* or *general*: an email of the former type contains a personal name (e.g. *mario.rossi@u-hopper.com*), whereas behind an email of the latter type is not a specific person, but for instance the office of a company (e.g. *sales@u-hopper.com*). The assumption made in this thesis, on which the development of Datatect is based, is that the email addresses are correct. This means that the incorrectness of a contact depends solely on the contents of the Name and Surname cells; thus, the total number of cells on which Datatect can act on is $2N$.

In general, Datatect deals with three versions of the same dataset:

`dirty dataset` **(D):** the input dataset of Datatect, the one that needs to be cleaned;

`cleaned dataset` **(C):** the output dataset cleaned by Datatect;

`ground truth dataset` **(GT):** the 100% correct version of `dirty dataset`.

In the relational algebra formalism, C, D and GT are three relations with attributes (Name, Surname, Email), and $|D| = |C| = |GT| = N$. A contact is identified by his Email across the three relations,

since their projection on Email attribute is the same $\Pi_{Email}(D) = \Pi_{Email}(C) = \Pi_{Email}(GT)$[1]. An ideal correct dataset has no empty cells when contacts have nominative emails, but both Name and Surname cells empty in case of general email type. The `ground truth dataset` respects this vision. Note that Name and Surname cells can contain one or more words.

As discussed in *Section* 2.1, Data Quality (DQ) is a multidimensional concept, and its evaluation passes through the assessment of each of its dimensions. There are three dimensions on which Datatect can act: accuracy, completeness and consistency. Therefore, these are the only DQ characteristics that are significant for this thesis, and on which the following discussion is based.

The methodologies for evaluating these DQ measures are not the same. All measures of completeness and consistency presented in *Section* 2.1 can be evaluated from the dataset whose level of DQ is to be estimated (*i.e.* the `dirty dataset`). In fact, to implement completeness measures, it is sufficient to calculate the number of empty cells in relation to the total (per attribute, per record, in the overall dataset). For consistency measures, it is possible to define certain rules *a priori* (e.g. there can be no numbers in a Name attribute cell).

However, to define the semantic and syntactic accuracy of the data it is necessary to know the correct value of the content of a cell. This means that it is impossible to evaluate the accuracy dimension without having a dataset containing the expected correct values in addition to the dataset to be evaluated: the `ground truth dataset`.

3.2 KPI for Datatect

The methodology used to evaluate the performance of the various versions of Datatect during its development is based on a comparison between the `cleaned dataset` and the `ground truth dataset`, and allows to evaluate the DQ in the three dimensions of accuracy, consistency, and completeness. This includes the use of a synthetic dataset, created specifically for this purpose. This dataset has been created together with its `ground truth dataset` version, so that the accuracy can also be calculated; the full explanation of its creation can be found in *Chapter* 4.

Datatect can perform three actions: fill empty cells, modify the contents of a cell, and leave the contents of a cell unchanged. A fill occurs when the cell in the `dirty dataset` is empty, and becomes non-empty in the `cleaned dataset`. A modification occurs when a cell is non-empty in both `dirty dataset` and `cleaned dataset`, but the content has changed; is considered a modification also the case where the content gets deleted (see *Table* 3.1). The last action is performed when the content of a cell remains unchanged from the `dirty dataset` to the `cleaned dataset`. Moreover, while cleaning the contacts, Datatect looks for those having general emails, emptying their Name and Surname cells.

At a first level, to access the performance of a certain version of Datatect one can evaluate its accuracy $\mathcal{A}$, *i.e.* the number of correct cells over total. In order for accuracy to better reflect Datatect's real performance, correctness of a cell was not calculated with respect to the `ground truth dataset` but rather to the maximum amount of information that was possible to extract (i.e. *Maximum Extractable Information*, as explained below). This is the main KPI used to track the performances each time a change in Datatect was made.

[1] Assuming there are no duplicate emails in the dataset.

Input	Output	Type of action
NaN	*Alfredo*	fill
Alfredo	*Alfred*	modification
Alfredo	*NaN*	
NaN	*NaN*	left unchanged
Alfredo	*Alfredo*	

Table 3.1: Example of Datatect's action on a Name cell. Input represents the cell's content in the `dirty dataset`, output is the content in `cleaned dataset`.

3.2.1 Maximum Extractable Information

The *Maximum Extractable Information* is defined as the amount of information contained in the `ground truth dataset` (named *Total Information*) that is in principle possible to pull out from the `dirty dataset`, without any additional knowledge. This has two components: the amount of information extractable from the email address only, and the information added by the content of the cells in the `dirty dataset`.

Imagine you have a `dirty dataset` where the columns Name and Surname are empty. In this scenario, all the information that Datatect can use to fill the cells has to come from the Email column. By comparing this column with the Name and Surname columns of the `ground truth dataset`, it is possible to know whether the entire contents of the cell can be retrieved or not. In particular, considering the Name column only:

Case 1: the email contains all the names recorded in Name. In this scenario, the recovery is possible, and the *Maximum Extractable Information* coincides with the *Total Information*.

Case 2: the email contains one but not all the names recorded in Name. Here, the *Maximum Extractable Information* is only the name contained in the email.

Case 3: the email is not easily traceable to the names recorded in Name, for example because it is formed by a nickname, or contains only the initials. In this case the *Maximum Extractable Information* is zero.

Note that "email content" is referred to any part of the email address, both the local part and the domain.

On the other hand, if the `dirty dataset` is non-empty, the Name and Surname columns contain additional information compared to Email column. This can bring the *Maximum Extractable Information* closer to the *Total Information*. In fact, if a Name cell has the same content in the `dirty dataset` and in the `ground truth dataset`, the *Maximum Extractable Information* coincides with the *Total Information* for that cell. The information in the `dirty dataset` can also be an integration with respect to the amount of information extractable from the email only: if a contact has two Names, but the email address contains only one of the two, the remaining one can be correctly identified if the Name of the `dirty dataset` contains it (also in the case when one name is contained in the email address, and the other is the one in Name column).

A side note must be done regarding accents and capital letters: the local part of email addresses contains only lowercase Latin letters, and thus names containing accents or special characters are flattened. For comparison purposes, the names in the two versions are considered equivalent.

3.3 Families of metrics

To get a more complete view of Datatect's performance, second-level metrics have been defined to evaluate DQ in the three dimensions of accuracy, completeness, and consistency. In this section, these metrics are presented, categorised by family.

Family 1

Entity-based metrics: how many entities (*i.e.* single words) Datatect is able to identify? See an example of this in *Table* 3.2.

Percentage of entities correctly found: the percentage of the sum over all the records of the number of entities that are correctly identified over the total number of entities in the `ground truth dataset`, without caring whether they are classified as Name or Surname.

Percentage of entities correctly found, Name attribute: the percentage of the sum over all the records of the number of entities that are correctly identified in Name over the total number of entities in the `ground truth dataset`, considering the Name attribute.

Percentage of entities correctly found, Surname attribute: the percentage of the sum over all the records of the number of entities that are correctly identified in Surname over the total number of entities in the `ground truth dataset`, considering the Surname attribute.

Name GT	Surname GT	Name C	Surname C
Anna Maria	Giacomelli	Anna	Maria Giacomelli
Sandra	Deanesi		Deanesi

Table 3.2: Mock dataset with two records: comparison between the ground truth (GT) and the cleaned dataset (C). For the first record, the number of entities recognised is 3 out of 3, in the second record, 1 out of 2. The names (surnames) correctly found and assigned are 1 (1) for the first record, 0 (1) for the second record.

Family 2

Metrics that evaluate the ability of Datatect to correctly target the cells that need a fill, a modification, or a "leave unchanged" action (see the example in *Table* 3.3).

Percentage of correctly targeted to fill: the percentage of cells on which a fill was made from those that needed it.

$$F = \frac{\sum_k |\Pi_{Email}(\sigma_{k\neq\emptyset}(C) \bowtie \sigma_{k=\emptyset}(D) \bowtie \sigma_{k\neq\emptyset}(GT))|}{\sum_k |\Pi_{Email}(\sigma_{k=\emptyset}(D) \bowtie_{Email} \sigma_{k\neq\emptyset}(GT))|}, \quad k = Name, Surname$$

Percentage of correctly targeted to modify: the percentage of cells on which a change was made from those that needed it.

$$M = \frac{\sum_k |\Pi_{Email}(\sigma_{k.D\neq k.GT \wedge k.GT\neq\emptyset} D \bowtie_{Email} GT) \cap \Pi_{Email}(\sigma_{k.C\neq k.D \wedge k.D\neq\emptyset} C \bowtie_{Email} D)|}{\sum_k |\Pi_{Email}(\sigma_{k.D\neq k.GT \wedge k.GT\neq\emptyset} D \bowtie_{Email} GT)|},$$

$$k = Name, Surname$$

Percentage of correctly targeted to left unchanged: the percentage of cells on which no action was taken among those that were already correct in the `dirty dataset` (and thus were to remain unchanged).

$$U = \frac{\sum_k |\Pi_{k,Email}(C) \cap \Pi_{k,Email}(D) \cap \Pi_{k,Email}(GT)|}{\sum_k |\Pi_{k,Email}(D) \cap \Pi_{k,Email}(GT)|}, \quad k = Name, Surname$$

Name GT	Surname GT	Name D	Surname D	Name C	Surname C
Anna Maria	Giacomelli		Maria Giacomelli	Anna	Maria Giacomelli
Sandra	Deanesi	Deanesi	Deanesi		Deanesi

Table 3.3: Mock dataset with two records and two attributes (Name, Surname): comparison between the `ground truth dataset` (GT), the `dirty dataset` (D) and the `cleaned dataset` (C). The cell correctly targeted to fill is 1, the cell correctly targeted to modify is 1, the cell correctly targeted to left unchanged is 1.

Family 3

Metrics answering the question: among all the targeted cells, how many correct actions were performed? *I.e.*, how many correctly filled cells are there among those that have been filled? Note that the correctly filled cells is a subset of the cells that have been filled, and that the case of a correct filling among the cells that were wrongly targeted to fill is not possible. The same is valid for modifications and left unchanged actions.

Percentage of correct filled over all the filled cells: the percentage of correctly filled cells over those filled.

$$F_{corr} = \frac{\sum_k |\Pi_{Email}(\sigma_{k.C=k.GT} C \bowtie GT) \cap \Pi_{Email}(\sigma_{k\neq\emptyset}(C) \bowtie \sigma_{k=\emptyset}(D) \bowtie \sigma_{k\neq\emptyset}(GT))|}{\sum_k |\Pi_{Email}(\sigma_{k\neq\emptyset}(C) \bowtie \sigma_{k=\emptyset}(D) \bowtie \sigma_{k\neq\emptyset}(GT))|},$$

$$k = Name, Surname$$

Percentage of correct modified over all the modified cells: the percentage of correctly modified cells over those modified.

$$M_{corr} = \frac{\sum_k |a \cap \Pi_{Email}(\sigma_{k.D\neq k.GT \wedge k.GT\neq\emptyset} D \bowtie_{Email} GT) \cap \Pi_{Email}(\sigma_{k.C\neq k.D \wedge k.D\neq\emptyset} C \bowtie_{Email} D)|}{\sum_k |\Pi_{Email}(\sigma_{k.D\neq k.GT \wedge k.GT\neq\emptyset} D \bowtie_{Email} GT) \cap \Pi_{Email}(\sigma_{k.C\neq k.D \wedge k.D\neq\emptyset} C \bowtie_{Email} D)|},$$

$$k = Name, Surname\,, \quad a = \Pi_{Email}(\sigma_{k.C=k.GT} C \bowtie GT)$$

Percentage of correct unchanged over all the unchanged cells: the percentage of cells that are correct and remained unchanged over those that have not been modified. By definition of the "leave unchanged" action, it is 1.

$$U_{corr} = \frac{\sum_k |\Pi_{Email}(\sigma_{k.C=k.GT} C \bowtie GT) \cap \Pi_{k,Email}(C) \cap \Pi_{k,Email}(D) \cap \Pi_{k,Email}(GT)|}{\sum_k |\Pi_{k,Email}(C) \cap \Pi_{k,Email}(D) \cap \Pi_{k,Email}(GT)|},$$

$$k = Name, Surname$$

3.3.1 Confusion matrices

For a better understanding of the behaviour of Datatect, confusion matrices related to the abilities of targeting, entity classification, and general email detection have been defined. The first one is evaluated separately for the Name and Surname attributes, and for the whole dataset; the last two only for the whole dataset.

in the following, the confusion-matrix is intended to have the true (expected) values as rows, and the predicted ones as columns.

Targeting

An error of Datatect may be due to a misidentification of the action to be performed on the cell, or to the erroneous result of a correct action. The presence of a correct cell in the `cleaned dataset` necessarily passes through a correct action identification: it is not possible to have in output a correct cell as a result of a misidentification.

In this scenario, measuring the Datatect's ability of targeting, *i.e.* the ability to correctly identify the action to be performed on a certain cell, is crucial. The confusion-matrix $M = (m_{ij})_{i,j\in[1,3]}$ associated to this ability is a $3x3$ matrix, where 3 are the actions that can be performed (fill, index 1; modify, index 2; leave unchanged, index 3). Note that, due to the construction of Datatect, $m_{1,2} = m_{2,1} = 0$: because of how the actions were defined, the it is not possible for a cell that needs a modification to be filled, and vice versa.

The total number of correct cells in the `cleaned dataset` is less than or equal to the sum of the diagonal terms $\sum_i m_{ii}$: the two coincide when none of the outputs' actions are erroneous. As a complement to the measures of the targeting ability, also the two components of the diagonal terms are evaluated: the number of the correct and incorrect cells resulting from the application of the correct action. Note that, due to the construction of Datatect, the number of cells that are correct after being left unchanged, if that was the action required, is zero. Thus, for the "leave unchanged" action, the diagonal term $m_{3,3}$ is equal to the number of correct cells resulting from that action.

Entity classification

This confusion matrix is intended to identify any imbalance in the classification of entities in the two name-surname categories. It is evaluated among all the entities correctly recognised by Datatect . Since this is a binary-classification (class 1: name; class 2: surname), the corresponding confusion-matrix $M = (m_{ij})_{i,j\in[1,2]}$ is a $2x2$ matrix. Note that the sizes of the two classes do not match perfectly, but they are comparable.

General email detection

The last confusion-matrix concerns the ability of Datatect to recognize whether a contact has nominative or general email. This is a binary classification, thus the corresponding confusion-matrix $M = (m_{ij})_{i,j\in[1,2]}$ is a $2x2$ matrix. Unlike the entity classification case, here the two classes (class 1: nominative; class 2: general) are unbalanced: typically the ratio between the sizes of the two is far from 1. Moreover, the intent is different: here the matrix should help in limiting the presence of false-negatives, *i.e.* the classification of a nominative email as general. In fact, due to the construction of Datatect, the identification of a general email ends the cleaning process for that record.

3.4 Data Quality of a user dataset

When a user uploads a dataset to Datatect, there is no way of knowing what the expected correct values are for that dataset: the uploaded dataset is the `dirty dataset`, and Datatect is responsible for creating its `clean dataset` version. Thus, in this case the evaluation of Data Quality does not include the dimension of accuracy, but only those of completeness and consistency. The methodology used is based on comparing the dataset loaded by the user with the same dataset cleaned by the cleaning method of interest (e.g Datatect).

Measures of record ($Com_1(i)$, where i is the email that identifies the contact), attribute ($Com_2(n)$ and $Com_2(s)$ for Name and Surname attribute respectively) and overall completeness (Com_5) have been defined for the dataset $R = D, C$ as:

- $Com_1^R(i) = \frac{1}{2}(|\sigma_{Email=i \wedge Name \neq \emptyset}(R)| + |\sigma_{Email=i \wedge Surname \neq \emptyset}(R)|)$;
- $Com_2^R(n) = \frac{1}{|R|}|\sigma_{Name \neq \emptyset}(R)|$, and similarly $Com_2^R(n)$;
- $Com_5^R = \frac{1}{2\,|R|}(|\sigma_{Name \neq \emptyset}(R)| + |\sigma_{Surname \neq \emptyset}(R)|)$.

The measure of consistency Con_6 is based on the rule that Name and Surname attributes can not contain non-letter characters (punctuation marks or numbers), and the first letter of each entity should be a capital letter[2]. If $\phi(X)$ is a function returning *True* when X respects the rule, and *False* otherwise, the measure of overall consistency (Con_6) for the dataset R is

$$Con_6^C = \frac{|\sigma_{\phi(Name) \wedge Name \neq \emptyset}(R)| + |\sigma_{\phi(Surname) \wedge Surname \neq \emptyset}(R)|}{|\sigma_{Name \neq \emptyset}(R)| + |\sigma_{Surname \neq \emptyset}(R)|}.$$

The estimate of Data Quality DQ for the dataset R is based on the measures of overall consistency Con_6 and overall completeness Com_5:

$$DQ^R = \frac{|\sigma_{\phi(Name) \wedge Name \neq \emptyset}(R)| + |\sigma_{\phi(Surname) \wedge Surname \neq \emptyset}(R)|}{2\,|R|}.$$

Eventually, the estimate of Data Quality increase ΔDQ has be defined as the difference between the `clean dataset` and the `dirty dataset` in terms of Data Quality:

$$\Delta DQ = DQ^C - DQ^D.$$

$\Delta DQ \in [-1, 1]$; a positive value of ΔDQ indicates that the cleaning method has increased the data quality of the user dataset.

[2] of course there are exceptions: the name *Mary-Michelle* contains a hyphen, or the surname *de Marco* has a lowercase initial; but these are considered to be negligible.

Chapter 4

Synthetic dataset

The synthetic dataset is fundamental in the development process: once set up, this is the *fil rouge* connecting all the improvements of Datatect. Its complete and correct version is the `synthetic ground truth dataset`, from which is created its `synthetic dirty dataset` version by removing information and introducing errors in the cells.

The synthetic dataset should be a realistic reproduction of a real world CRM dataset. Its creation and dirtying faces some challenges, on whose overcoming depends the reliability of the KPIs. First of all, there is no single typical user of CRM systems: one company can use it to track contacts of sales or marketing offices, another one can store lead information coming from social media or email campaigns. The difference here is in the emails: the first CRM dataset likely has mostly company emails, which usually contain plain name and surname in the local part; the latter instead will have a lot of personal emails, whose local part is sometimes a nickname. In addition to that, due to privacy issues is difficult to find a publicly available CRM dataset from which extract information on its composition: of course names, surnames and email addresses are sensitive data and can not be disclosed. Eventually, the creation of a synthetic dataset implies the generation of fake contacts (containing name, surname and email address). Here there is a risk of bias to take into account: the names and surnames cannot be picked up from the same dataset used in Datatect, and fake email addresses should not resemble the same patterns that Datatect considers and from which it can extract information.

There are two main sources of knowledge on CRM data composition: two internal datsets, which have no privacy issues and whose data can therefore be examined; and the datasets of datumo users, subject to the GDPR regulations and of which only statistical data have been retained. The approach taken starts with obtaining from the internal datasets information on missing values, types and frequency of errors, presence of contacts having general email, and then validating these data on the datasets of datumo users. This information is then used to construct the synthetic dataset.

This chapter is organized as follows. In the first two sections, the analysis of the internal and users $\mathcal{U}$ datasets is presented. Afterwards, a critical analysis of the synthetic dataset developed together with datumo is shown, in order to answer the question: why do we need another synthetic dataset? The last two sections concern the creation of two versions of the syntethic dataset $\mathcal{S}$: the `synthetic ground truth dataset` $\mathcal{S}^{GT}$ and the `synthetic dirty dataset` $\mathcal{S}^{D}$.

4.1 Internal datasets

The two internal datasets have size[1] of $S = 1464$ and $S = 1210$, respectively. In the following, they are considered unified as "internal dataset" $\mathcal{I}$ with size $S = 2674$. This dataset is provided with both ground truth $\mathcal{I}^{GT}$ and dirty $\mathcal{I}^{D}$ version.

4.1.1 Ground truth internal dataset

The contacts with nominative email are 92% of the total: the internal nominative dataset $\mathcal{I}_{nominative}$ has size $S = 2460$. The analysis of the top-level domains shows that 54.23% of the contacts has a country specific top-level domain, for a total of 52 different codes; the distribution of occurrence is shown in *Figure* 4.1. The 58.34% of the emails with a country code top-level domain display the code *it* (*Italy*): this is explained by the provenance of $\mathcal{I}$, which is used in an Italian context.

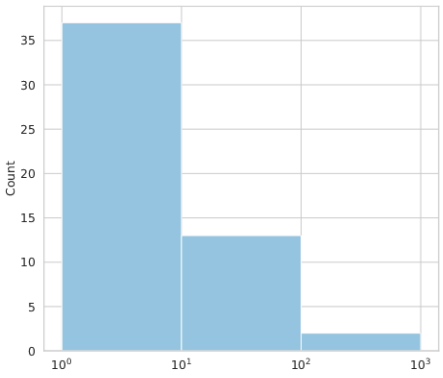


Figure 4.1: Country codes top-level domains per occurrence. The two country codes appearing more than 100 times are *it* (846) and *eu* (131).

Regarding the domains of email addresses, the 11.29% present a free email provider domain: thus 34.48% of the contacts have an email address which is not traceable to a specific country, but neither is provided by a free email provider.

The names and surnames recorded in $\mathcal{I}^{GT}_{nominative}$ can be composed by one or more entities; single-entity names and surnames are respectively the 89.60% and the 86.54% of the cells. The difficulty of dealing with multiple-entities names and surnames is that the entities do not always have the same weight: although the full name is Gerardo Andreas, the person may use only the first name in most situations. This imbalance in entities increases the complexity of the dataset in terms of the ease of establishing a link between email address and name and surname cell contents. To be realistic, a synthetic dataset must respect this complexity.

In an attempt to find recurring patterns in the email address local part, name and surname cell content has been compared with the email address. The patterns considered are a composition of name, surname, and their initials; as shown in *Figure* 4.2, 62.03% of the nominative emails are of the kind {*name*}.{*surname*}, where the dot indicates the presence of one or more separators (dot, hyphen, underscore, digits), and {*name*} is the person name (properly processed by removing accents and making it lowercase). The patterns considered, that cover the 92.11% of the email addresses, are the following:

1. {*name*}.{*surname*}
2. {*surname*}.{*name*}
3. {*surname*}
4. {*name*}
5. {*name*}{*surname*}
6. {*surname*}{*name*}
7. {*n*}.{*surname*}
8. {*s*}.{*name*}
9. {*surname*}.{*n*}
10. {*name*}.{*s*}
11. {*n*}{*surname*}
12. {*s*}{*name*}
13. {*surname*}{*n*}
14. {*name*}{*s*}
15. {*n*}{*s*}
16. {*s*}{*n*}
17. {*n*}.{*s*}
18. {*s*}.{*n*}

[1] the number of contacts in the dataset.

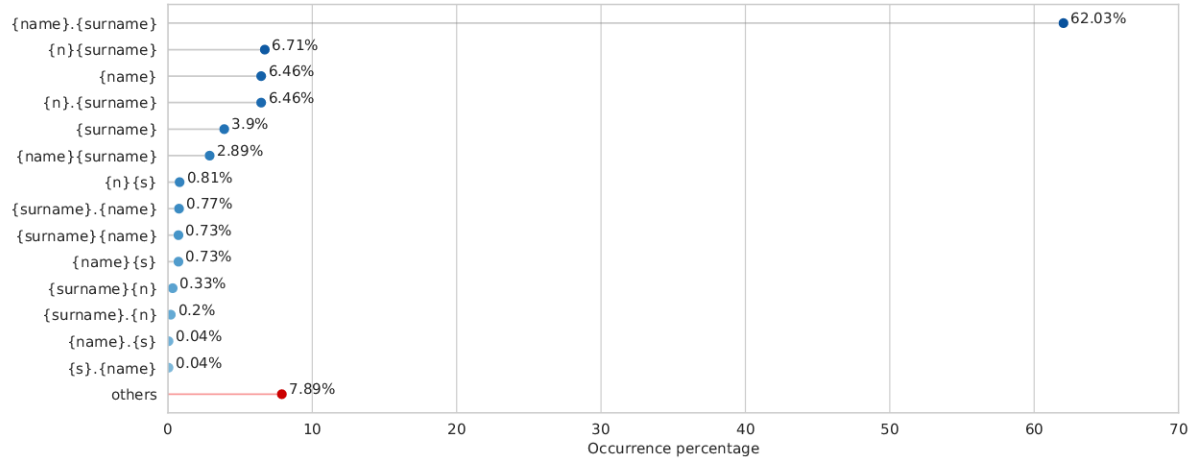


Figure 4.2: Percentage of occurrence of the email patterns found in $\mathcal{I}_{nominative}$. The dot indicates the presence of one or more separators: dot, hyphen, underscore, digits. The initials are denoted by {s} (initial of the surname), {n} (initial of the name). "others" indicates the percentage of email addresses which can not be traced to any pattern (among those considered).

The dot indicates the presence of one or more separators: dot, hyphen, underscore, digits. The initials are denoted by $\{s\}$ (initial of the surname), $\{s\}$ (initial of the name), while with $\{name\}$ and $\{surname\}$ are indicated the full names and surnames.

In the case of single-entity names, $\{name\}$ and $\{n\}$ are self explanatory (and the same applies to surnames). The tricky part comes with multiple-entities names and surnames, where some variability is introduced. When fully present in the local-part (*i.e.* not as initials), name (or surname) with multiple entities is labelled as

separable, when the entities are all present in the local part, and are separated from each other with one or more separators;

non separable, when the entities are all present in the local part, but without any separator between them;

only first, when only the first entity is present in the local part;

only last, when only the last entity is present in the local part.

On the other side, when only initials are present in the local-part, name (or surname) with multiple entities is labelled as

all initials, when the entities are all present in the local part with their initials;

first initial, when only the first entity is present in the local part with his initial;

last initial, when only the last entity is present in the local part with his initial.

This classification affects the pattern recognition. As an example, consider a contact with name "Anna Maria" and surname "Rossi": $\{name\}$ change according to the label associated to the name, and can be "anna.maria", "annamaria", "anna" or "maria" based on the label. Also the initials change: $\{n\}$ can be "am", "a" or "m". This classification leads to the fact that in all the local parts *anna.maria.rossi*, *annamaria.rossi*, *anna.rossi* and *maria.rossi* is recognized the pattern

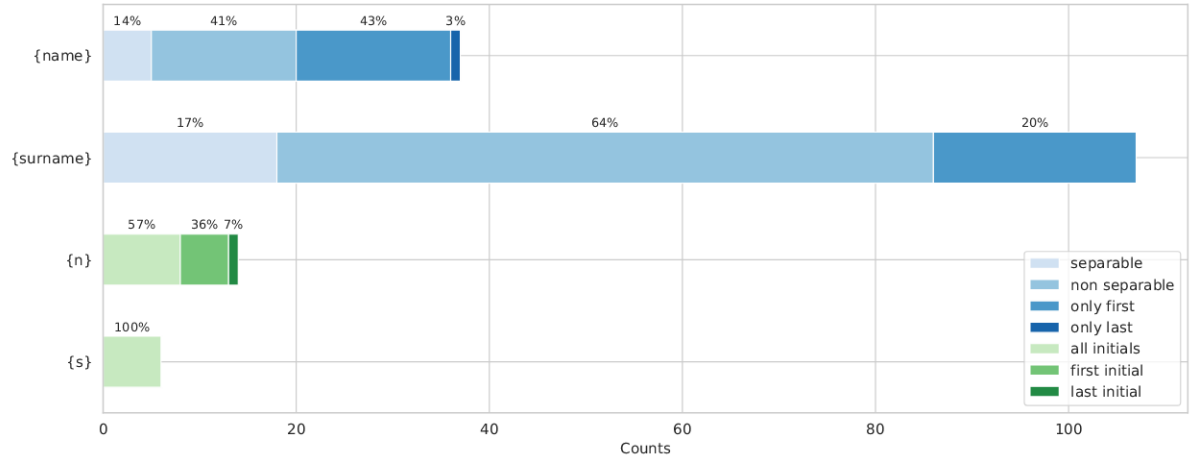


Figure 4.3: Barplot representing how multiple-entities names and surnames enter email patterns in $\mathcal{I}_{nominative}$. In blue, the number of contacts with multiple-entities names (surnames) whose email address contains $\{name\}$, divided into "separable", "non separable", "only first" and "only last". In green,the number of contacts with multiple-entities names (surnames) whose email address contains the initials $\{n\}$, divided into "all initials", "first initial" and "last initial". The percentages indicate the weight of each label within each category.

$\{name\}.\{surname\}$.

In multiple-entities surnames, the 63.55% of the patterns containing $\{surname\}$ have a "non separable" surname in the local part of the email address. On the other hand, the most frequent labels for names are "only first" (43.24%) and "non separable" (40.54%). Note that the 19.05% of the multiple-entities names and the 16.91% of the multiple-entities surnames belong to contacts where no pattern was recognized in the local part of the email address. For further clarifications concerning the variability of multiple-entities names and surnames presence in email patterns, see *Figure* 4.3.

4.1.2 Dirty internal dataset

The correct contacts in the dirty internal nominative dataset $\mathcal{I}^D_{nominative}$ are 57.20%. Moreover, 64.47% of name cells and 59.96% of surname cells are already correct. Less than half of the cells actually need a cleaning action, the others simply need not be changed because they are already correct: for this reason, the approach to cleaning must be very conservative, not assuming that the content of a cell is incorrect.

The presence of empty cells in the internal dataset is not negligible: 16.38% of contacts have both name and surname cells empty. More specifically, 16.38% of name cells and 28.23% of surname cells are empty. Note that there is no contact with the name cell empty, while the surname is non-empty. It should also be noted that the emptiness of a cell does not necessarily indicate lack of information: in fact, it may happen that the content of the name cell is moved to the surname cell, and vice versa.

In $\mathcal{I}_{nominative}$, the percentage of empty name cells is 14.27%, while the same value for surname cells is 25.33%.

To gain insights into the amount of information actually contained into name and surname cells of $\mathcal{I}^D_{nominative}$, a comparison has been performed between entities in the union of name and surname cells in $\mathcal{I}^D_{nominative}$ and in $\mathcal{I}^{GT}_{nominative}$. This comparison is carried out after a standardization of the content of the cells: punctuation removal and entity lowercase. The results are shown in *Figure* 4.4.

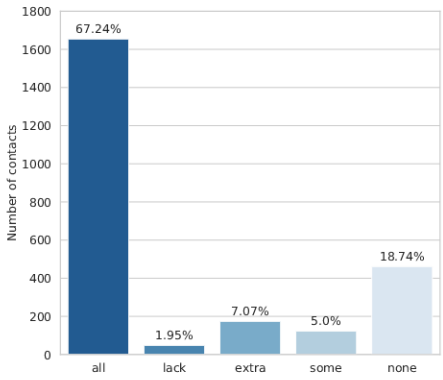


Figure 4.4: Amount of information contained in the union of name and surname cells of $\mathcal{I}^D_{nominative}$. With *all* are indicated those contacts where the entities are all and only the correct ones, at most exchanged between the attributes name and surname; *lack* indicates those contacts where there is a lack of information: all the entities in the dirty contact are correct, but some are missing; *extra* means that all the correct entities are present in the dirty contact, but there are also extra (not correct); *some* indicates those contacts where at least one entity of the dirty contact is correct, but not all of them (*i.e.* there is simultaneously lack and extra information); *none* indicates those contacts where none of the entities in the dirty contact is correct.

Errors in $\mathcal{I}_{nominative}$ are of various kind. First of all, the 7.85% of the name cells contains at least one character which is not a letter and neither a hyphen, and this number drops to 2.11% in the surname cells. This errors are mainly comments in parenthesis or after a comma, name initials with dots, or the local part or the full email address pasted into the name/surname cell.

Secondly, there are typos errors in the cell content. In particular, based on an estimation performed with Levenshtein distance[2], it comes out that 1.10% of the contacts contain a typo. Is considered a typo when the Levenshtein distance between an entity of the dirty contact and the corresponding ground truth is 1 or 2.

A recurrent error is also the presence in name or surname cells of the name of the company: 7.20% of contacts contain the second-level or third-level domain names in the attributes name or surname.

4.2 Datasets of datumo users

Each of the 19 different datumo users (registered before 12/11/2022) was able to try it several times, thus producing multiple versions of the `clean dataset` (*i.e.* the dataset resulting from datumo action). These datasets differ in both number of boosts enabled, size (some contacts may be added/dropped between versions), and version of datumo that worked on them.

The data considered for the analysis are collected from 6 datasets that meet the following requirements:

1. email attribute without empty values: Datatect's action starts from email attribute, and therefore when this is empty, no action on name or surname attributes can be performed;
2. processed by datumo 0.24.0 (latest version);
3. `clean dataset` version with all the possible boosts enabled, so as to ensure an even comparison;
4. largest possible size among all the versions of cleaned dataset, to have the larger number of data as possible;
5. dataset size $S \geq 10$.

[2]Levenshtein distance between two strings is the minimum number of single-character edits (insertions, deletions or substitutions) required to change one string into the other.

In *Table* 4.1 the sizes of the datasets considered are shown. Note that the difference between them is huge: the smallest dataset considered has $S = 23$, the largest has $S = 3379$. Because of this, the values extrapolated from the datasets of datumo users are obtained from an unified dataset of size $S = 10219$, called "users dataset" $\mathcal{U}$. This dataset is provided with only the dirty version $\mathcal{U}^D$, and with the version cleaned by datumo $\mathcal{U}^C$.

dataset ID	Size
1	23
2	3318
3	2113
5	1198
6	188
7	3379

Table 4.1: Size of the six datasets considered.

A starting point is to extrapolate the number of non-empty cells, attribute-wise and total (name and surname attributes). In *Figure* 4.5(a) are shown the distributions of the percentage of non-empty cells out of total number of cells in name and surname attributes of the original dataset, and the distribution of the percentage of non-empty cells out of total number of cells in the overall dataset. In *Figure* 4.5(b) the percentages of the number of non-empty cells out of the total number of cells evaluated for $\mathcal{U}^D$ are reported. From this values it is possible to notice that, unlike the internal dataset, the number of empty name cells is almost equal to the number of empty surname cells.

In *Figure* 4.6(a) are shown the distributions of the percentages of modifications performed out of the total number of non-empty cells, *i.e.* the non-empty cells in $\mathcal{U}^D$ whose content were somehow modified. In *Figure* 4.6(b) the percentages of the number of modified cells out of total number of non-empty cells evaluated for $\mathcal{U}^D$ are reported.

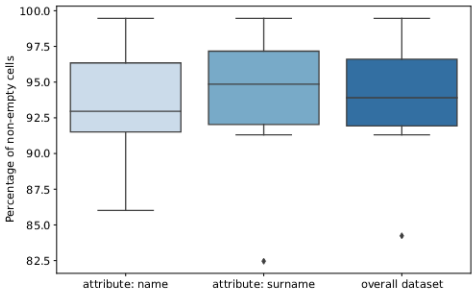
These values represent only the number of cells that were originally non-empty and on which datumo acted on, and are used to obtain an estimate of the number of non-empty but incorrect cells in $\mathcal{U}^D$. In fact, a modification performed by datumo can also incorrectly modify the content of a cell, when it was correct at the origin.

In *Figure* 4.7 are reported the confusion matrices for the performances of datumo (version $0.24.0$) on non-empty cells, for attributes name, surname and overall. These values are evaluated on the internal dataset, and should give an estimate of datumo's ability to modify incorrect cells, leaving the others unchanged.

As a result of the comparison between the values in *Figure* 4.7 and the ones reported in *Figure* 4.5(b), by means of a proportion it is possible to estimate at 10.48% (6.97%) the percentage of name (surname) cells that need a modification being incorrect; the overall percentage of cells that need a modification is 8.71%.

The estimations of incorrect cells are always higher than the values of cells modified by datumo: this seems reasonable, under the hypothesis that datumo has been developed to edit cells carefully. Although, considering that the internal dataset has been a basis in the development of datumo, here there is a probable underestimate of the number of originally correct cells, that were modified by his action.

Eventually, the presence of contacts with general email was considered: 8.61% of the emails in $\mathcal{U}^D$ are found by datumo as general. Comparing this value with the performance of datumo in general email detection (*Figure* 4.8), the percentage of general emails in $\mathcal{I}$ is estimated at 12.21%.

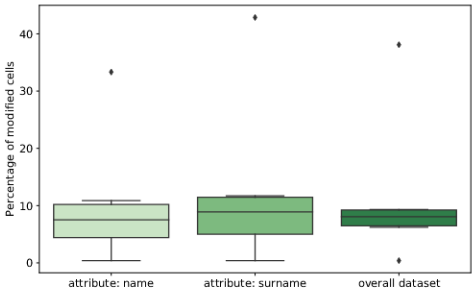


(a)

	Percentage
name	91.87%
surname	91.37%
overall	91.62%

(b)

Figure 4.5: (a) Distribution of the percentage of non-empty cells out of total number of cells in $\mathcal{U}^D$ (attributes: name, surname, overall dataset); (b) percentage of non empty-cells out of total number of cells in $\mathcal{U}^D$.



(a)

	Percentage
name	6.87%
surname	5.43%
overall	6.15%

(b)

Figure 4.6: (a) Distribution of the percentage of modified cells out of total number of non-empty cells in $\mathcal{U}^D$ (attributes: name, surname, overall dataset); (b) percentage of non empty-cells out of total number of non-empty cells in $\mathcal{U}^D$.

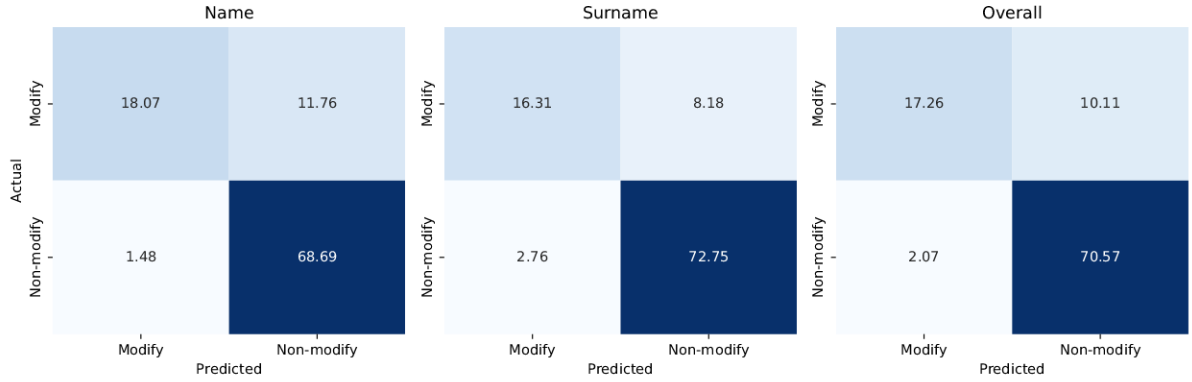


Figure 4.7: Confusion matrices for the modifications performed by datumo on in $\mathcal{J}^D$; the values are the percentages evaluated over the non-empty cells of in $\mathcal{J}^D$.

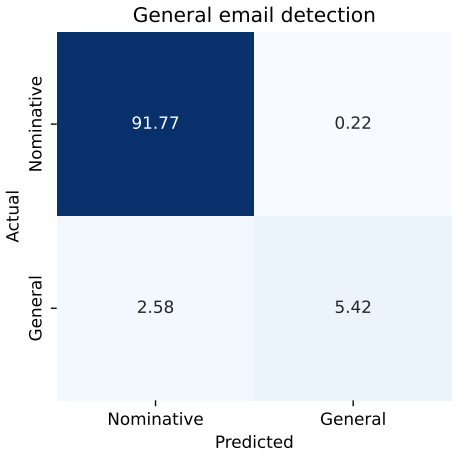


Figure 4.8: Confusion matrix for general email detection performed by datumo on $\mathcal{I}^D$; the values are the percentages evaluated over the total number of contacts of $\mathcal{I}^D$.

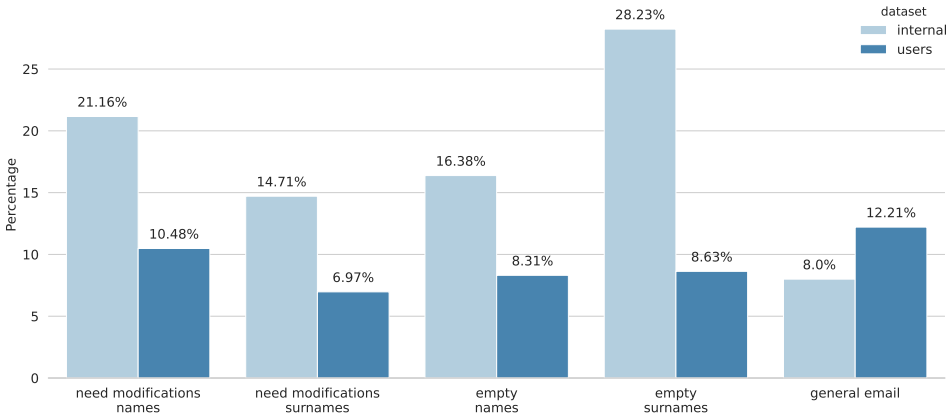


Figure 4.9: From left to right, for both $\mathcal{I}$ and $\mathcal{U}$: percentage of name cells that are non-empty and incorrect ("need modifications"), percentage of surname cells that are non-empty and incorrect ("need modifications"), percentage of name cells that are empty, percentage of surname cells that are empty, percentage of contacts with general email address.

4.2.1 datumo users vs internal datasets

Although little information can be extracted from $\mathcal{U}$, it can nevertheless be used to get an idea of how the characteristics of $\mathcal{I}$, and consequently the data obtained in *Section* 4.1, are representative of an average CRM dataset. In *Figure* 4.9 are shown the values found for the cells that need a modification, the cells that are empty, and the contacts with nominative email in both $\mathcal{I}$ and $\mathcal{U}$.

Except for the percentage of contacts with general email, all values in $\mathcal{U}$ are lower than the ones in $\mathcal{I}$. In particular, the percentage of empty surnames in $\mathcal{U}$ cells is less than $1/3$ of that in $\mathcal{I}$: respectively 8.63% and 28.23%. This is the major difference; the values of empty names, and names and surnames that need a modification have a difference of 7 to 11 percentage points.

Despite some values retrieved from $\mathcal{U}$ are estimated using datumo's performance on $\mathcal{I}$, values from both information sources were averaged in order to build the synthetic dataset $\mathcal{S}$. In *Table* 4.2 are reported the means for each case.

non-empty and incorrect cells		empty cells		general email contacts
names	surnames	names	surnames	
15.82%	10.84%	12.35%	18.43%	10.11%

Table 4.2: Summary of the analysis of the internal and the users datasets.

4.3 Another synthetic dataset?

At the beginning of datumo development process, there was little CRM data to rely on. For this reason, the synthetic dataset developed together with datumo does not perfectly match a real-world dataset composition. One of the first issues concerns the lack of consideration of contacts with general email: as shown in the previous sections, a CRM dataset contains a non-negligible percentage of email addresses (and thus contacts) not necessarily traceable to an individual. In the case of the internal datasets, this percentage is around 8%. In addition to that, the nations considered for fake contact generation are only five: Italy, Germany, Great Britain, Spain, France.

The creation of fake email addresses is based only on eight patterns for the local part, that cover only 72.86% of $\mathbb{J}$ nominative email addresses:

1. {name}.{surname}@domain, 54.21%;
2. {name}{surname}@domain, 1.95%;
3. {surname}.{name}@domain, 0.57%;
4. {surname}{name}@domain, 0.61%;
5. {surname}@domain, 3.62%;
6. {name initials}.{surname}@domain, 5.69%;
7. {surname}{name initials}@domain, 0.2%;
8. {name initials}{surname}@domain, 6.01%.

This synthetic dataset considers only those cases where the name is formed by one or two single entities (same for the surname); and in case of two, in the email address they are joined by an underscore: the email of Anna Maria Rossi, in the first pattern, becomes anna_maria.rossi@domain. This of course reflects only part of the case history, and the underscore facilitates the two names extraction: if the email address was annamaria.rossi@domain, it would be more difficult to trace it back to the name Anna Maria.

Eventually, there are issues concerning the dirtying of the synthetic dataset, which is performed by means of two actions: name and surname swap, and content erasure. Specifically, the records affected by swaps are the 3.37% of the total contacts with nominative email, and among them:

1. simple swap between name and surname, 90.36%;
2. name added after surname (as surname), not found;
3. name added before surname (as surname), not found;
4. surname added after name (as name), 9.64%;
5. surname added before name (as name), not found.

The contacts affected by content erasure (and which do not fall into the swaps case) are 25.03% of the total contacts with nominative email address. Out of them, the 58.77% present empty name and empty surname, and 41.23% of them miss the surname.

Considering that the amount of correct contacts having nominative email in $\mathcal{I}$ is 55.95%, 15.64% remains out, representing the contacts with wrong content, but not due to an erasure or a swap.

A new synthetic dataset is created with the aim of increasing the adherence of fake data to real data, both in terms of creating more truthful contacts (adding more email patterns, nationalities, ..) and in terms of dirtying process (reducing that 15.64% remaining out).

4.4 Synthetic Dataset creation

In this section the process by which the `synthetic ground truth dataset` $\mathcal{S}^{GT}$ was built is explained.

Fake names, surnames and email addresses are needed for the creation of the synthetic dataset. The sources of those data must be independent with respect to those used in Datatect, so to avoid the introduction of a bias. The fake names and surnames for the contacts with nominative email are generated from NameDataset [41] (see *Section* 2.4), randomly sampling from the top 10^3 names (half male, half female) and surnames for a specific country. Each of the countries considered[3] contribute to the synthetic dataset with the same number of contacts.

In the creation of Name and Surname attributes for the fake contacts, statistics with respect to multiple-entities and single-entity names and surnames (*Section* 4.1) were considered. The noise in NameDataset makes these attributes more realistic, allowing language discrepancy between entities forming name and surname.

With regard to email addresses, their creation was split between domain and local part creation. Email address domains are considered to be of two types: coming from free email providers, or containing a company or institution name. The list of free email providers used include *gmail.com*, *yahoo.com*, *hotmail.com*, *outlook.com*. The domains based on company names are obtained from Wikidata, extracting them from the URL of the official page of 10^4 enterprises founded after 1950. However, in the latter case the top-level domain has been modified to meet the requirements in terms of country-codes and non country-codes top-level domains. Country-codes top-level domains were assigned regardless of the country of the attributes Name and Surname, so as to ensure some noise: not all the contacts with tld *.it* have Italian names, and so on.

Local part creation is based on the patterns found in *Section* 4.1, and has been created with the attributes Name and Surname for each contact. Following the intuition that the local part patterns tend to be the same within the same company, a link between local part and domain has also been established (when the latter is company-based). When assigning the domain to a contact a, the same domain is assigned to n next contacts among the m remaining ones displaying the same pattern as a. n is is randomly selected from the uniform distribution $U(0, min\{10, m\})$. This method allows the distribution of patterns to remain unchanged, while increasing the complexity of $\mathcal{S}$. Note that this allows also the presence of different patterns for the same company domain: the company domains are randomly chosen from a list without replacement, meaning that the same company domain can be chosen multiple times independently of the procedure shown above.

[3] Alpha-2 country codes: MT, MX, BR, DK, US, IE, PL, BE, AT, EE, ID, CN, DE, CA, IT, NL, LT, PT, MY, CL, UY, TR, IN, HR, CO, IL, RS, TW, NO, SI, GB, ZA, FR, JP, RU, FI, GR, CH, AE, ES, HU, SE, CY.

The creation of contacts with general email address must be considered separately. Name and Surname attributes were created empty, considering that no name or surname can be associated to a general email address. The local part of the email address is randomly chosen from a set created thanks to real general email addresses available on *hunter.io*, so that they are as realistic as possible.

The synthetic dataset has been created with 10^4 contacts, 10.11% of which have general email according to *Table* 4.2. The dataset generation is not fully deterministic, although some parameters remain fixed: the percentage of contacts with nominative email (*Table* 4.2), the pattern ratios of the nominative email addresses (*Figure* 4.2), the compositions of multiple-entities names and surnames (*Figure* 4.3), as well as the percentage of free email providers and country-codes top-level domains.

4.5 Synthetic Dataset dirtying

In this section the process by which the `synthetic dirty dataset` $\mathcal{S}^D$ was built is explained. This is carried out by dirtying the contacts of the `synthetic ground truth dataset` $\mathcal{S}^{GT}$ just created.

The dirtying process affects Name and Surname attributes, while Email remains unchanged. The number of empty cells, as well as the number of cells that should be already correct, are taken from *Table* 4.2. The remaining contacts having nominative email are dirtied by exploiting entities information collected in *Figure* 4.4. The process has a non-deterministic component, but each realization of the process retains the characteristics listed above.
The assumption made is that 100% of the empty Name cells belong to contacts with also an empty Surname cell, and that 95% of the correct Surname cells belong to fully correct contacts. Thus, to recover the values in *Table* 4.2, the number of Surname cells needing to be emptied e_s and the number of Name/Surname cells which should remain unchanged u_n/u_s are tracked. The dirtying operations are performed sequentially on separate subsets of nominative-type contacts after the removal of contacts with both attributes correct or both empty, and are listed below.

Simple attribute swap The content of Name and Surname cell is swapped, regardless of whether they consist of one or more entities each.

Complex attribute swap If both Name and Surname consist of a single entity and $e_s > 0$, both are put in Name attribute, while Surname is emptied. If both Name and Surname consist of a single entity but $e_s \leq 0$, a simple swap occurs. Else, if $u_s > 0$ and Name is formed by more than one entity, entities in Name are shuffled. Finally, if none of the above occurs, half entities are put in Name and half in Surname, randomly.

Entity erasure If both Name and Surname consist of a single entity, and $u_s > 0$, Name attribute is filled with the Surname. Else, if both Name and Surname consist of a single entity, and $e_s > 0$, Surname is emptied and Name is filled with Surname according to $u_n > 0$. Else, if both Name and Surname consist of a single entity but none of the above occurs, a simple swap occurs. Finally, if none of the above occurs, half entities are randomly put in Name and half in Surname, after removing one of them.

Entity insertion If $u_n > 0$, comments are added to Surname attribute. Else if $u_s > 0$, comments are added to Name attribute. Finally, if none of the above occurs, Surname is added to Name attribute, and Surname attribute is replaced with domain.

Entity erasure and insertion If $u_n > 0$, with probability $p = 0.5$ a typo in inserted in Surname attribute, or with probability $p = 0.5$ Surname attribute is filled with the initial only; the typo consists

in the removal of a single letter, in any but the first position within the entity. Else, if $u_s > 0$ the same as before occurs, with Name instead of Surname attribute. Finally, if none of the above occurs, Surname is filled with Name attribute, and in Name is inserted a title such as *Dr*, *Mr*, *Ing*.

Entity total erasure If e_s, Surname is emptied and with equal probability a typo is inserted in Name, Name is replaced with its initial, Name is replaced with the local-part of the email. Else if e_n, the same as above occurs. Finally, if none of the above occurs, Name is replaced with its initial and in Surname is inserted a typo.

In addition to this, with a probability $p = 0.05$ one entity in Name or in Surname attribute of the contact is doubled at an initial stage. The dirtying process for contacts with general email includes the insertion of domain and/or local part from the email address in Name and/or Surname attributes.

Chapter 5

Datatect

Datatect operates on one contact at a time, cleaning the Name and Surname attributes by integrating them with the information contained in the email address[1]. Contacts are extracted from the `dirty dataset` and then, once cleaned by Datatect, are reassembled to form the `clean dataset`. The cleaning process was designed to remain as faithful as possible to the information already present in the Name and Surname cells, according to the assumption (validated on available real data *Chapter* 4) that the contacts in a CRM dataset are mostly already correct. In addition, the presence of general email addresses is also to be taken into account as these records do not have the attributes Name and Surname (or, if they have it, it cannot be traced back to email in any way).

Datatect consists of several blocks responsible for different actions, the main ones being: general email detection, email-attribute integration, attribute assignment. These blocks are sequential, and the exit points are only after the first block (when the email of a contact is identified as general) or after the last one (see *Figure* 5.1). The actions performed in each block are based on tasks such as information extraction from the local-part of the email, information extraction from the attributes, language detection, typos correction, entity classification in name or surname, and word segmentation.

The first part of this chapter is devoted to each of the tasks, and in the last section is presented the structure of Datatect based on the blocks of general email detection, email-attribute integration, attribute assignment.

A note should be done regarding the data underlying the tasks performed by Datatect: since the source is always the same, the Wikidata corpus, the errors due to the presence of spurious names and

[1] An email address is of the form {local_part}@{domain}.

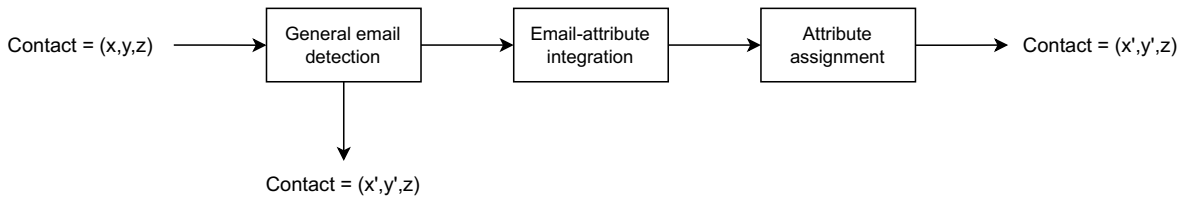


Figure 5.1: Structure of Datatect.

surnames (*i.e.* entities recorded as names but being surnames, or belonging to an incorrect language dataset, or with a spelling error) are propagated through all the tasks.

5.1 Attribute preprocessing

CRM datasets can have consistency issues, as it is defined in *Chapter* 3: the contents of Name and Surname cells can be altered by the presence of characters that are not letters, such as numbers or punctuation marks, or by the fact that rules such as capital letters for proper names are not observed. In addition, it often happens that content is added that is not a name or surname, but a company name, role, or title. All this makes the extraction of relevant information more difficult.

Consider a contact whose Name attribute is "dr. Hoffmann marta (U-HOPPER)". The relevant information to be extracted is *Hoffmann Marta*, but the cell is polluted with the indication of a title, with a comment in brackets referring to the company, with a lowercase name. Note that at this stage "relevant information" deals only with the identification of proper names or surnames, without considering that they may be in the wrong attribute cell or in the wrong order.

Attribute preprocessing is responsible for extracting the relevant information from the contents of the Name and Surname cells. This is essential both to make an initial cleaning of the contact and to make sure that it is comparable with the information contained in the email. The preprocessing steps are as follows, executed sequentially.

1. Comments identification and removal: a comment is identified thanks to regular expressions as a sequence of characters in brackets or after a hyphen.
2. Domain and email removal: an entity is removed if it matches the the email or the email domain (top-level domain excluded).
3. Punctuation marks and numbers removal: dots, commas, semicolons, digits are removed.
4. Remove honorifics and titles, performed with an external list including words such as *dr*, *mrs* etc.
5. Remove duplicated entities, both of the same attribute or across attributes.
6. Extra spaces removal.
7. Capitalization of initials of each entity, lowercase the other letters.

To make attributes comparable with emails, further operations were performed. Email addresses do not have accents, or apostrophes: after lower-casing the letters, any accent, diacritic mark and apostrophe is removed (*Josè* becomes *jose*, *Dell'Angelo* becomes dellangelo). In this way the so-called *plain name* and *plain surname* are obtained, which contain the relevant information extrapolated from the cells Name and Surname.

5.2 Local-part preprocessing

This task is complementary to the attribute preprocessing, and is responsible for extracting as much information as possible from the local-part of the email address, taken by itself. This means that, without any external contribution, the entities forming the local-part are identified and isolated by exploiting the presence of separators (any non-letter character). In this way, from the local part *mary.anne* it is possible to extract two entities: mary and anne.

Entities extracted from the local part are of various types. In some cases, there are no separators within an entity, although they should: it is the case of *maryanne*. In others, the entity extracted is only one or two letter long, because it is the initial of a name or surname, or both. Many such cases could be added, but what must be emphasised is that the information extracted from the local part at this stage (in the form of a *list of entities*) is further processed later, for instance using word segmentation techniques.

5.3 Typos correction

The detection and correction of typing errors is made possible by comparing entities extracted from the local part with plain names and surnames. The assumption made is that the entities coming from the local part are always correct: if a contact has *mara* as plain name, and *marta@u-hopper.com* as email address, the correct Name attribute should be "Marta". However, identifying the two entities to be compared is far from easy, as the extraction of entities from the local part is not straightforward (e.g. *martar@u-hopper.com*).

The detection of a typo is done by means of the Levenshtein distance between two strings a, b $lev(a, b)$, defined as the minimum number of single-character edits (insertions, deletions or substitutions) required to change one string into the other[43]. Given two entities a and b, where a comes from the email and b comes from the plain name or surname, an error is found if

$$1 \leq lev(a, b) \leq 2.$$

To limit the uncertainty about the correspondence between an entity in the local part and one in the plain name/surname, some constraints have been added, such as that the first letter of both strings must coincide $a[0] = b[0]$.

5.4 Word segmentation

As previously mentioned in *Chapter* 2, word segmentation in Datatect is applied to the local part of email addresses. It is indeed very common for it to contain the name and the surname of the contact without any separator between them (e.g. *johngreen*), and it is therefore necessary to develop a method to separate them. This word segmentation approach is based on a dictionary where names and surnames are related to their occurrence (the Wikidata dataset, see *Section* 2.4). Each word w is associated with a probability $P(w) = o_w/N$, where o_w is the number of occurrences of the word w and $N = \sum_w o_w$ is the total number of occurrences of the words in the dictionary. These probabilities are the basis of the model, as the segmentation of the input string occurs when the product of the single substrings is maximised.

It is not possible to think of a dictionary containing all existing names and surnames; for this reason, a word not present in the dictionary cannot be immediately discarded by associating a null probability with it. A small but non-zero probability should always be associated to a candidate segmentation, even if it contains words not found in the dictionary. In fact, a human being could easily say that the string *mianhussain* is more likely to consist of the substrings *mian* and *hussain* rather than of *mia*, *nhussa* and *in*, even without directly knowing the names *Mian* and *Hussain* (as if they did not belong to his dictionary): a motivation may be that a two-letter surname seems to us to be less likely than a seven-letter one.

In order to overcome these two issues, a probability distribution for unknown words $P^l(u)$ was defined from the distribution of lengths l of names and surnames in the Wikidata dataset[2]. A single probability distribution is obtained by averaging the probability distributions of names $P_n^l(u)$ and surnames $P_s^l(u)$:

$$P^l(u) = \frac{P_n^l(u) + P_s^l(u)}{2}.$$

The two have been considered separately and then joint to reflect the assumption that an unknown word resulting from a candidate segmentation is just as likely to be a name or a surname. The number of names and surnames in the Wikidata dataset does not coincide, and therefore had to be normalised separately before merging them. To keep the probability associated with a word present in the dictionary higher than that of a not-present word, the distribution of the unknown words has to be reduced by a factor $F \cdot P^l(u)$. The value $F = \frac{1}{10}N^{-2}$ was chosen as the optimal trade-off on the basis of several tests.

The challenge of local part word segmentation is made even more difficult by the presence of one or two letters, usually initials, attached to the name you wish to isolate: some examples of this are *mariaa*, *josemgarcia*, *geraldrm*. Local parts are corrupted by the presence of non-names and non-surnames between entities: to consider this aspect as well, the probability of an unknown word of length 1 is set to $P = 1/N$, as the less probable word in the dictionary.

To summarise, the probability $\mathcal{P}$ associated to each segment s in the candidate segmentation is

$$\mathcal{P}(s) = \begin{cases} o_s/N & \text{if } s \in \mathcal{D}, \\ 1/N & \text{if } s \notin \mathcal{D} \wedge len(s) = 1, \\ 10^{-1}N^{-2}P^l(s) & \text{otherwise,} \end{cases}$$

where $\mathcal{D}$ is the set of dictionary keys, o_s is the number of occurrences of the segment, P^l is the unknown words distribution.

One final consideration: languages. The assumption that the distribution of lengths of names and surnames is the same for each language is excessively strong, since the evidence on the Wikidata dataset shows the opposite (as it is possible to see from the single distributions in *Appendix* A).

Moreover, consider the names *Hannah* and *Hanna*: if we try to segment the local part *hannahwilliams*, probably we would find [*hannah*, *williams*] as best candidate; on the other hand, if the local part is *hannahnowak*, the best candidate should be [*hanna*, *h*, *nowak*]. Why? Because *Hannah*, as *Williams*, are popular names and surnames in the US, while *Hanna* and *Nowak* are popular in Poland, suggesting that the remaining *h* is likely to be the initial of a second name (for example, *Helena*). This example shows that performing local part segmentation on an unique dictionary for all languages can be misleading.

The last two considerations made support the choice of using language-specific dictionaries to perform local part segmentation in Datatect.

5.5 Entity classification in name and surname

The incorrectness of a contact is often due to the fact that the Name attribute also contains surnames, and vice versa. It is therefore necessary to develop a method to classify entities as names and surnames

[2] In the dataset, a word appears a number o_w of times.

(*i.e.* a binary classification), so that they can be assigned to the correct attribute.This task is difficult because of the slight differences that sometimes exist between names and surnames (e.g. *Daniele* and *Danieli* are an Italian name and an Italian surname, respectively), and because of the presence of patronymics, which in not all languages, and not always, are preceded by particles such as the Italian "di" (e.g. *Martino* is both a name and a surname in Italian).

A first way of performing this task is to classify the entity according to its presence in a dictionary of names or surnames. This method leaves the attribution of many entities doubtful, when an entity that does not appear in either dictionary or is present in both.

A second approach is to use a machine or deep learning models for performing this classification, exploiting the features selection techniques used for language identification (character bigrams and trigrams, see *Section* 5.6) as input to a SVM or artificial neural network. However, these models have not proved very effective in practice: the shortness of names and surnames, combined with the difficulties listed at the beginning of this section, prevent them from achieving sufficient accuracy on the Wikidata dataset[3]. In addition to this, performance are further deteriorated when switching to real data, since those used for training are only a small subset of them (the Wikidata dataset contains only a few names and surnames, with respect to the vast variety of names and surname in the real world).

As shown later in *Chapter* 6, the performances of Datatect on the synthetic dataset are better if the entity classification is performed with the dictionary approach. The motivation can be summarised as follows: given the high probability that an entity is already correctly classified (see *Chapter* 4), it is preferable that a shift from the Name attribute to the Surname only takes place when it is (almost) certain that that entity is actually a surname, and vice versa.The benefit of the doubt is given to entities that are not in the dictionary, thus limiting the risk of moving an entity when it was already in the correct attribute. To further reduce this risk, dictionaries by language should be used instead of the one obtained from the overall Wikidata dataset.

5.6 Language identification

Performing entity classification and word segmentation on a one big names dataset is not efficient: as shown in the previous sections, a one-language approach limits the risk of error in both cases. Thus there is the need of a filtering procedure that somehow permits to access to a language subset of that one big names dataset.

One approach may be to use the top-level domains of emails. However, this presents a number of problems, starting from the fact that just a few records have an email with a country code top-level domain *Chapter* 4. Moreover, in the cases where a country code top-level domain is present, there is no guarantee that this matches the nationality of the contact: this is for example the case of an employee named John Smith who works for an Italian company (and whose email top-level domain is "it"). On top of that, a country-specific dataset can be noisier than one language-specific for the purposes of this thesis: some countries have more than one official languages (Belgium, Canada..), and some languages are spread over multiple countries (German is spoken both in Germany and in Austria).

[3]The distribution of balanced accuracy for each language model has a mean and standard deviation of 0.78 and 0.04, respectively. The best model used to calculate balanced accuracy is an SVM, the kernel of which changes according to the language considered (*linear* for Chinese and Turkish, *rbf* in the other cases).

For these reasons there is a need for a different approach that allows each record to access only a language specific dataset: identification of the contact language, understood as the language from which his name originates.

As introduced in *Chapter* 2, the model used for language identification is a Support Vector Machine, with character bigrams and trigrams as features. The model has been trained on the Full names corpus presented in *Section* 2.4: 335584 full names among 13 countries (see *Table* 2.2). Each full name has been preprocessed making it lowercase, removing punctuation and diacritical marks, removing numbers and roman numbers. Then, full names that contained a spurious entity, *i.e.* a name or surname that should belong to the dataset of another language, were removed from each language dataset. Given N_e^α the occurrences of entity e in the language α dataset, full names containing e in language α dataset are removed if

$$\exists\, \beta \neq \alpha \quad | \quad \frac{N_e^\alpha}{N_e^\beta} < \frac{1}{10} \quad \forall\, \beta \in \mathcal{L},$$

where $\mathcal{L}$ is the languages set. After this operation, the sizes of the datasets are on average 92% of the originals[4].

Bigrams and trigrams were obtained from each string, after adding spaces at the beginning and at the end so that prefixes and suffixes are counted appropriately. Principal Component Analysis has been performed to reduce the feature set size to 400, with 74.25% of total variance explained.

This is a multi-class classification problem: each input is labelled with one of the 13 languages. Training has been performed on 85% of each language dataset, thus maintaining the same ratio of train and test samples for each language. SVM with a linear kernel is the best model found among those tested[5], reaching a balanced accuracy[6] of 0.84 on the test set. The confusion matrix is shown in *Appendix* B.

Datatect uses this trained model to detect the language of each contact. The input is the string consisting of plain name, plain surname and the list of entities of the local part. The output is one of the 13 languages considered.

5.7 Datatect structure

This section presents the overall process of cleaning a contact, from its entry into Datatect until its exit as clean contact.

5.7.1 General email detection

This section presents the first block constituting Datatect, in which the email of a contact is classified as nominative or general. At the end of this block, if the contact email is labelled as nominative it continues into the next block (the email-attribute integration block), otherwise it exits the Datatect. This block is made necessary by the fact that a contact having a general email should not have Name and Surname attributes, and therefore its cleaning process should be different. The assumption made

[4] ar: 94%, de: 93%, el: 91%, en: 88%, es: 89%, fr: 87%, it: 92%, ja: 97%, nl: 90%, pt: 91%, ru: 94%, tr: 97%, zh: 88%.

[5] SVM with rbf kernel, feed-forward neural network

[6] Balanced accuracy is the average of sensitivity (true positives rate) and specificity (true negatives rate) for each class, then averaged over k classes [44].

in this thesis is that contacts having general email are correctly cleaned when Name and Surname attributes are empty: the motivation is that a general email often is the email of an office (and so there is no single individual behind[7]).

The purpose of this block is to identify the contacts with general email, trying to limit the number of false positives (*i.e.* nominative that are labelled as general). This is achieved through the following steps, which are carried out sequentially until one of the statements is true.

1. Is the local part in the list of "alert words"[8]?
2. Is the contact email a disposable email address[9]?
3. The list of entities coming from the local part contains one word being in a list of common English and Italian nouns, but it is not in the Wikidata dataset?

If the answer to any of these questions is yes, the contact email is labelled as general and its Name and Surname attributes are emptied. Otherwise, the contact continues and enters the second block: email-attribute integration

5.7.2 Email-attribute integration

Email-attribute integration is the core of Datatect. It is the block in which the information extracted until now from the attributes (*i.e.* the plain name and surname) and from the local part (*i.e.* the list of entities) is compared and integrated. This is done starting from the list of entities $\mathcal{E}$, iterating on all its elements e_i and comparing each of them with the entities in the plain name $n_i \in \mathcal{N}$ and in the plain surname $s_i \in \mathcal{S}$.

The general idea is the following. If e_i is equal to an entity in plain name or surname, and if its length is greater than a threshold value of 10, word segmentation is applied to it and the attribute that contains it is updated. The hypothesis is that this entity e_i can be a non-separable part of the local part pasted in one attribute, but it is actually formed by two separated entities: take as an example a contact with Surname attribute "garciagonzalez" and an email *ana.garciagonzalez@u-hopper.com*.

Otherwise, if the entity e_i is not in plain name or in plain surname, the possible cause[10] can be one of the following.

1. There is a typo in one of the entities in plain name or in plain surname that causes the mismatch.
2. e_i is formed by two or more separate entities (e.g. *annarossi*); the single entities may be in $\mathcal{N}$ or in $\mathcal{S}$, or not.
3. e_i can not be traced back to any n_i or s_i; it is additional information.

The first inspection made by Datatect when the entity e_i is not found in plain name or surname is to check whether it is a typo. typo detection occurs if $lev(e_i, x_i) \in [1, 2]$ (*Section* 5.3), $x_i \in \mathcal{N} \cup \mathcal{S}$, with the constraint $e_i[0] = x_i[0]$ (same first letter) and the first or last part of e_i should not be equal to

[7] The case where a general email address has a person associated to it, perhaps the contact person within the office, is negligible.

[8] It is a list of words commonly used in working contexts, such as *info*, *office*, *support*, *reply* and so on.

[9] It is a temporary email address, identified thanks to a list of disposable email providers.

[10] Of course these are only the assumptions made, reality can be much more creative.

the initials of the entities in $\mathcal{N}$ or in $\mathcal{S}$. The reason in the following: if $\mathcal{N} = [anna]$ and $\mathcal{S} = [rossi]$, $e_i = annar$ is not a typo for "anna".

Then, Datatect considers the second hypothesis: if it is found any $x_i \in \mathcal{N} \cup \mathcal{S}$ that is contained in the entity e_i, the segmentation is straightforward (if $\mathcal{N} = [anna]$, the entity *annarossi* is divided into *anna* and *rossi*). On the other hand, when this does not happen, word segmentation (*Section* 5.4) is applied if the length of the entity e_i is greater than a threshold value t ($t = 5$ if e_i is the only entity in list of entities $\mathcal{E}$, else $t = 10$).

Lastly, the third option is considered: if the entity does not seem to be related to a typo, if it does not contain any $x_i \in \mathcal{N} \cup \mathcal{S}$ and if word segmentation can not be applied, the entity is added into the Name attribute if it is empty, else is added to Surname attribute.

5.7.3 Attribute assignment

The last block is dedicated to a rearrangement of the entities that are in the Name and Surname attributes; the purpose is to decide whether an entity is in the correct attribute or whether it should be moved. To do this the entity classification task (*Section* 5.5) is exploited.

The procedure is kept as conservative as possible, and starts by classifying each entity as "name", "surname" or "dubious". All entities already present remain in the Name attribute, provided they have not been classified as "surname"; entities classified as "name" but which were present in the Surname attribute are then added. The same is done for the Surname attribute. In this way, "dubious" entities are left in the attribute they were already in.

In the special case where the total number of entities is 2, and after the previous steps they are both in one attribute, they are divided equally between the attributes Name and Surname. This forcing is done following the idea that most contacts have only one name and one surname, but that it is often the case that they are not classified correctly. The results of the Datatect performance on the synthetic dataset show that this choice leads to a real improvement in the cleanliness of the contacts (see *Chapter* 6).

A final consideration must be made regarding surnames composed of a particle that is placed before the main entity: this is the case with surnames such as "de Marco", "von Habsburg", "De la Cruz". Those particles with their corresponding main entity are identified and inserted directly into the Surname attribute.

Attribute assignment is the last step: at the end of this block, the cleaned contact exits Datatect.

Chapter 6

Performance results

To evaluate the performance of Datatect, the metrics and evaluation methodologies presented in *Chapter* 3 were applied on the 10^4 contacts of the synthetic dataset $\mathcal{S}$ (*Chapter* 4). Furthermore, the latest version of datumo (0.24.0) was tested on $\mathcal{S}$ with the aim of enabling a comparison between datumo and Datatect. Data Quality increase ΔDQ for both cleaning methods (Datatect and datumo) has been calculated, and its agreement with the overall accuracy $\mathcal{A}$ is shown.

The first two sections are devoted to the performance evaluation based on the ground truth dataset $\mathcal{S}^{GT}$ of Datatect and datumo, respectively. In the last section ΔDQ between the dirty $\mathcal{S}^{D}$ and clean $\mathcal{S}^{C}$ version are shown, for both Datatect and datumo cleaning methods.

6.1 Datatect on synthetic dataset

In this section the results obtained by Datatect on the synthetic dataset $\mathcal{S}$ are reported. The main KPI (*i.e.* the overall accuracy $\mathcal{A}$) and the various metrics (see *Section* 3.3) were calculated with respect to the *Maximum Extractable Information* of $\mathcal{S}$. In *Table* 6.1 and in *Figure* 6.1 the results are shown, while the overall accuracy reached is

$$\mathcal{A} = 0.86 \pm 0.03.$$

Results show that Datatect is able to fill and modify the content of the cells without too much influence on the cells that are already correct: 91% of them remain untouched. Modifications and fills are performed with high accuracy: 79% of the cells that have been modified as needed, have been modified correctly; 88% of the cells that were empty, have been filled correctly.
For what concerns entity recognition and assignment, more than 95% of entities were correctly identified, and misclassified entities are around 7% for both Name and Surname attribute.
Results for general email detection show that less than 1% of the nominative emails were misclassified as general, which is an important desired result. However, there remains a difficulty in correctly classifying general emails.

As previously mentioned in *Chapter* 5, here the version of Datatect with an SVM-based entity attribution is evaluated. The purpose is to show how much the dictionary-based approach for entity classification in name and surname is better than the SVM-based approach. The overall accuracy of $\mathcal{A} = 0.75 \pm 0.04$ and the results (shown in *Figure* 6.2) demonstrate that the dictionary-based entity

Family 1	Entities correctly found	0.95 ± 0.02
	Names correctly found and assigned	0.91 ± 0.03
	Surnames correctly found and assigned	0.90 ± 0.03
Family 2	Cells correctly targeted to fill	0.95 ± 0.02
	Cells correctly targeted to modify	0.83 ± 0.04
	Cells correctly targeted to being left unchanged	0.91 ± 0.03
Family 3	Cells correctly filled over the filled	0.88 ± 0.03
	Cells correctly modified over the modified	0.79 ± 0.04

Table 6.1: Results of Datatect performance on the synthetic dataset maximum extractable information.

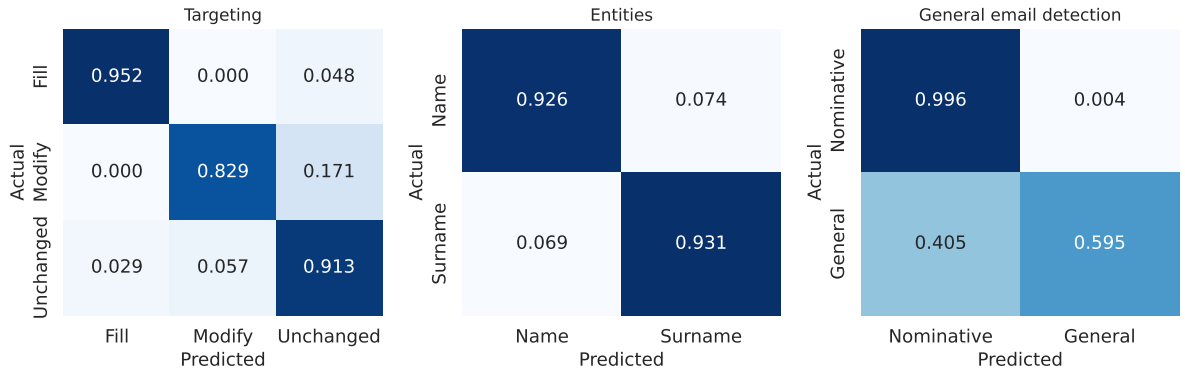
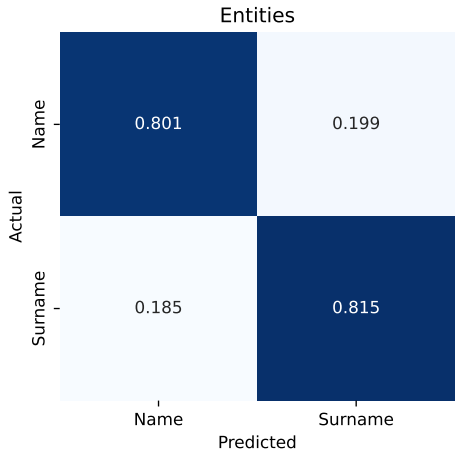


Figure 6.1: Confusion matrices for targeting the correct action (fill, modify, leave unchange), entities attribution and general email detection in Datatect on $\mathcal{S}$.



(a)

Entities correctly found	0.96 ± 0.02
Names correctly found and assigned	0.79 ± 0.04
Surnames correctly found and assigned	0.79 ± 0.04

(b)

Figure 6.2: (a) Confusion matrix for entities attribution in Datatect (version with SVM-based entity assignment) on $\mathcal{S}$. (b) Family of metrics 1, results of Datatect (version with SVM-based entity assignment) performance on the synthetic dataset maximum extractable information.

attribution is the choice that maximises Datatect performance. A lower overall accuracy is due to two factors: a targeting error, or an error in the outcome of the correct action. In this case, the entities found are the same, but the misattribution of some of them makes many cells incorrect, thus lowering the overall accuracy. In fact, the comparison between *Figure* 6.2(a) and *Figure* 6.1 shows that there is a drop in the percentage of names and surnames correctly attributed: the number of misclassified Names and Surnames is around 19% (instead of the 7% of the other version). This difference in the correct attribution of entities affects the overall performance, and results in this SVM-based approach for entity classification being discarded.

6.2 datumo on synthetic dataset

The results of datumo on the *Maximum Extractable Information* of $\mathcal{S}$ show an overall accuracy of

$$\mathcal{A} = 0.75 \pm 0.04.$$

Results (see *Table* 6.2 and *Figure* 6.3) show that datumo has a harder time keeping the contents of already corrected cells unchanged: 86% of them remain untouched, compared to 91% of Datatect. Regarding modifications and fillings, also here the results are worse, particularly in the case of the correct modifications: only 51% of the cells that have been modified as needed, have been modified correctly (compared to 79% of Datatect). Moreover, the percentage of already correct cells that get modified is higher in datumo than in Datatect (11% vs 6%), and the same is for the percentage of incorrect but non-empty cells that did not get modified (35% vs 17%). This can be summarised by stating that both in targeting the action to be performed and in performing the right action, Datatect performs better than datumo.
For what concerns entity recognition and assignment, the entities correctly identified are comparable, as well as the attribute classification (even if datumo makes fewer mistakes among names, and more among surnames).
Misclassified contacts with nominative email are slightly less (0.2% compared to 0.4%), while general emails classification is worse.

The dirty version of the synthetic dataset $\mathcal{S}^D$ has an overall accuracy of $\mathcal{A} = 0.71 \pm 0.05$. Both cleaning methods increase the accuracy of the dataset, although the improvement made by datumo is minimal ($+4\%$ of correct cells). This small improvement is attributable not so much to datumo's inability to find the correct entities (the number of entities found is almost the same for datumo and Datatect), but rather to a difficulty in integrating them correctly (classifying them as names and surnames, detecting possible errors, eliminating extra entities such as the domain). In addition, the number of cells originally correct that are dirtied by datumo has a great influence.

Results show that Datatect is both more conservative (because it dirties fewer cells that are originally correct) and more effective in modifying and filling than datumo. The difference in terms of overall accuracy reflects these findings. However, one has to keep in mind that the results obtained depend on the composition of the synthetic dataset: it is necessary to refine its construction by collecting further data from CRM systems before one can make this claim in an absolute sense.

6.3 ΔDQ in Datatect and datumo

Finally, ΔDQ for both Datatect and datumo was evaluated. In *Table* 6.3 results for Data Quality measured on $\mathcal{S}^D$, $\mathcal{S}^C$ and their difference ΔDQ for both Datatect and datumo (the measure on $\mathcal{S}^D$

Family 1	Entities correctly found	0.95 ± 0.02
	Names correctly found and assigned	0.91 ± 0.03
	Surnames correctly found and assigned	0.85 ± 0.04
Family 2	Cells correctly targeted to fill	0.86 ± 0.03
	Cells correctly targeted to modify	0.65 ± 0.05
	Cells correctly targeted to being left unchanged	0.86 ± 0.03
Family 3	Cells correctly filled over the filled	0.81 ± 0.04
	Cells correctly modified over the modified	0.51 ± 0.05

Table 6.2: Results of datumo performance on the synthetic dataset maximum extractable information.

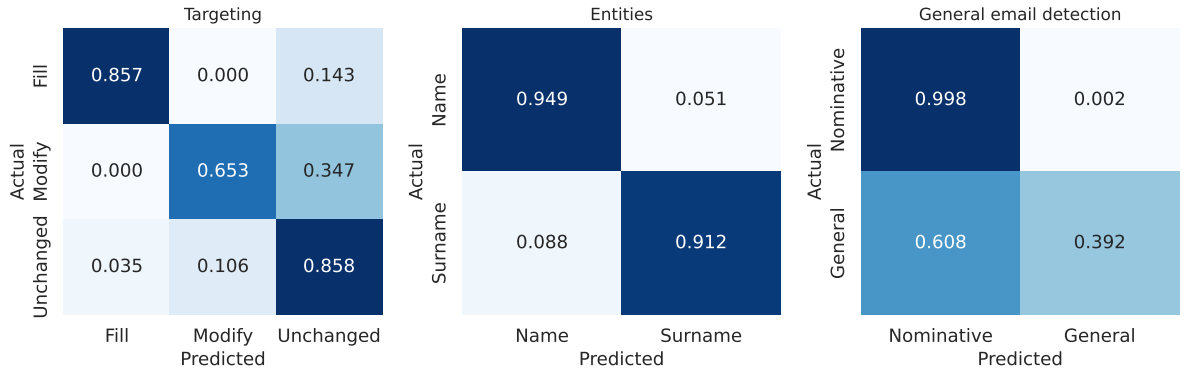


Figure 6.3: Confusion matrices for targeting the correct action (fill, modify, leave unchanged), entities attribution and general email detection in datumo on $\mathcal{S}$.

	DQ^D	DQ^C	ΔDQ
Datatect	0.69 ± 0.05	0.90 ± 0.03	0.20 ± 0.08
datumo	0.69 ± 0.05	0.86 ± 0.03	0.16 ± 0.08

Table 6.3: Data Quality measured on the synthetic dirty dataset DQ^D, on the cleaned dataset DQ^C, and data quality improvement ΔDQ for Datatect and datumo.

is the same in both cases, and is given only for completeness) are shown. In both cases there is an increase in the overall Data Quality, confirming the effectiveness of both methods in cleaning the given dataset.

According to this measure of Data Quality, Datatect performs better than datumo (has higher ΔDQ). However, this difference is mainly due to the larger number of cells filled by Datatect with respect to datumo, because other differences between the two methods (such as the correct entity attribution in Name or Surname, or the correctly modified cells) cannot be detected with this measure.

This measure proves to be consistent with the results shown in the previous sections, as the ΔDQ also suggests that Datatect increases the Data Quality of the dataset more than datumo.

Chapter 7

Conclusions

The aim of this thesis was to develop a method for cleaning a Customer Relationship Management (CRM) system datasets containing personal information, thereby increasing its Data Quality. This cleaning method should correct any errors in Name and Surname attributes of a contact in the dataset, also using information that can be retrieved from the email address (which is assumed to be correct). Furthermore, the method developed was to prove more robust than the currently existing method: datumo.

In this thesis Datatect, a method for cleaning personal information, was presented. It takes advantage of different techniques, such as rule-based approaches, machine learning algorithms, natural language processing methods. The structure of Datatect is divided into three main parts: initially, the type of contact is assessed in order to optimise the cleaning process; then the information in the Name and Surname attributes is supplemented with that extracted from the email address; and finally the attributes Name and Surname are reassigned and sorted. To evaluate Datatect performance, a synthetic dataset was created based on the analysis of real CRM system datasets.

Results show that the developed method is able to increase the Data Quality of a dataset from a CRM system, achieving an accuracy in correct cells of $\mathcal{A} = 0.86 \pm 0.03$. Moreover, Datatect proved to be more accurate in contact cleaning than datumo: the performance evaluated on the synthetic dataset are better for the former than for the latter (which achieves an accuracy of $\mathcal{A} = 0.75 \pm 0.04$). Second-level metrics assessing specific abilities (the ability to correctly fill, modify, leave unchanged the content of a cell) also confirm this result.

This thesis offers several hints for future developments. The various tasks performed by Datatect, such as language identification, name-surname classification, and word segmentation, can be further investigated. Moreover, other techniques and other approaches can be designed to clean a CRM system dataset, completely changing the structure of Datatect. However, their improvement certainly also depends on greater availability of name and surname data, and some effort must also be made in this direction. Future developments should also be based on the collection of more data from CRM systems, in order to update the synthetic dataset so that performance evaluations are increasingly reliable.

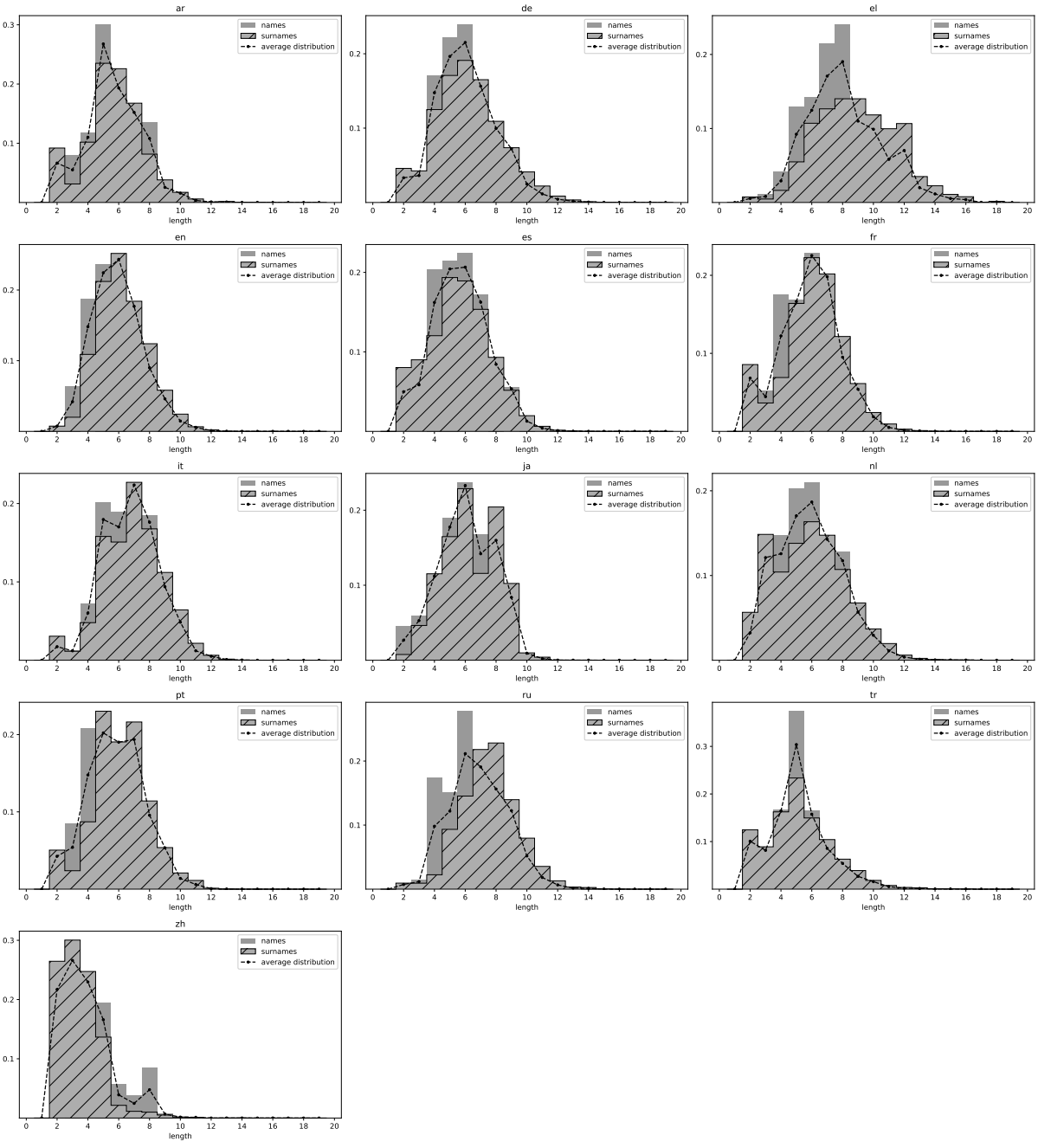
Datatect can be further extended to include other attributes, either by cleaning them or using them for information extraction. An address, for example, can be used for language identification; the email address can be also used to clean a "Company Name" attribute. This cleaning method can be

also extended to enrich the attributes of a CRM system dataset, for instance by guessing the gender of the contact.

In addition, other datasets containing personal information in a broad sense may benefit from this search. Although Datatect is based on email addresses, it is possible to modify it in such a way as to exploit alternative sources of information such as social security number and *codice fiscale*, perhaps to clean health-related datasets.

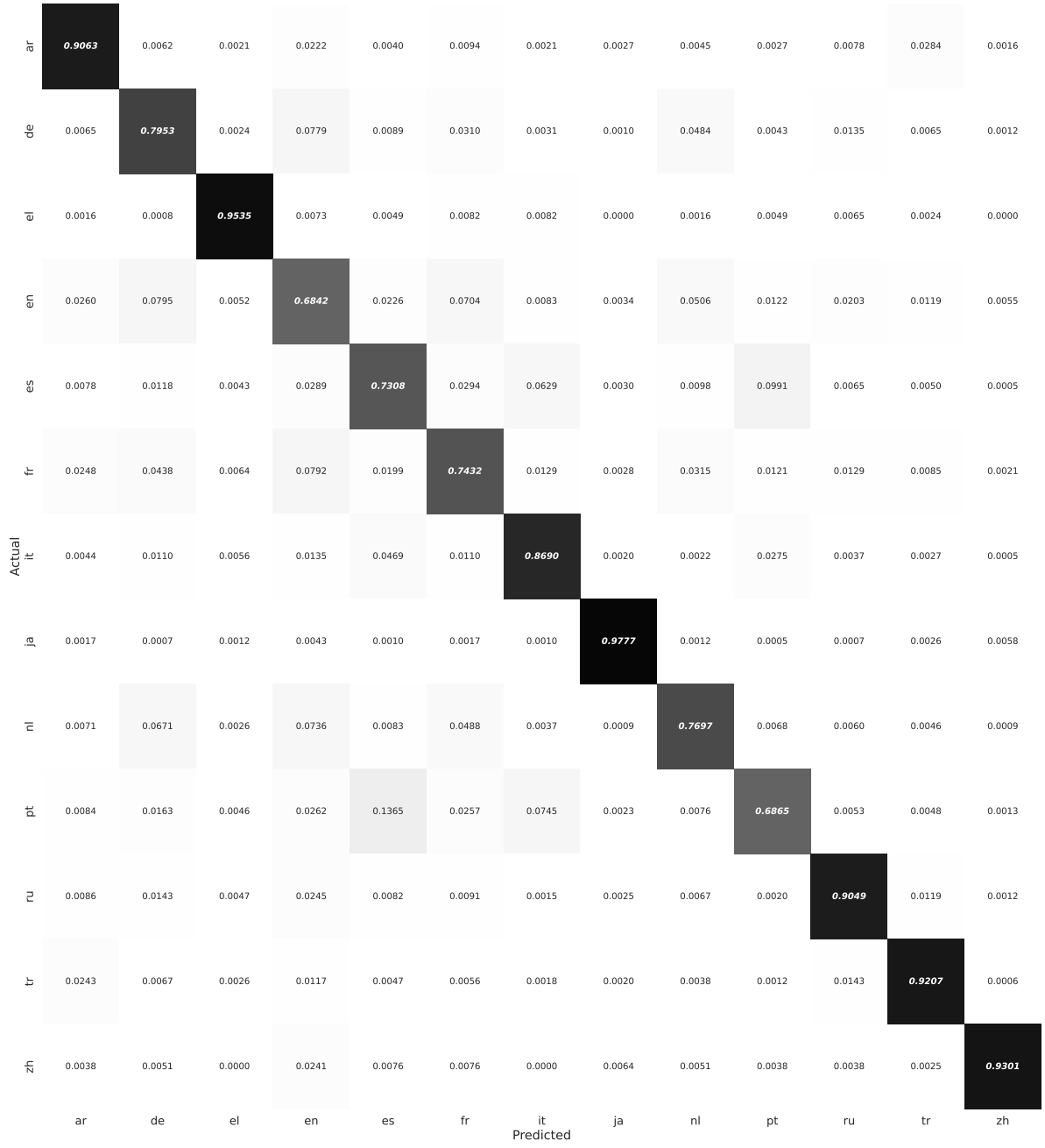
Appendix A

Distributions of names and surnames lengths, and their average distribution, for each language Wikidata dataset.



Appendix B

Confusion matrix for the language classification on the test set (Wikidata Full-names dataset).



Bibliography

[1] F. Buttle and S. Maklan, *Customer Relationship Management: Concepts and Technologies*. Routledge, 2015.

[2] R. Y. Wang and D. M. Strong, "Beyond accuracy: What data quality means to data consumers," *Journal of management information systems*, vol. 12, no. 4, pp. 5–33, 1996.

[3] I. O. for Standardization, *ISO/IEC 25024: 2015: Systems and Software Engineering-Systems and Software Quality Requirements and Evaluation (SQuaRE)-Measurement of Data Quality*. ISO/IEC, 2015.

[4] T. C. Redman, *Data quality for the information age*. Artech House, Inc., 1997.

[5] M. Jarke, M. A. Jeusfeld, C. Quix, and P. Vassiliadis, "Architecture and quality in data warehouses: An extended repository approach," *Information Systems*, vol. 24, no. 3, pp. 229–253, 1999.

[6] M. Bovee, R. P. Srivastava, and B. Mak, "A conceptual framework and belief-function approach to assessing overall information quality," *International journal of intelligent systems*, vol. 18, no. 1, pp. 51–74, 2003.

[7] I. O. for Standardization, *ISO/IEC 25012: Software Engineering: Software Product Quality Requirements and Evaluation (SQuaRE): Data Quality Model*. ISO/IEC, 2008.

[8] T. Jauhiainen, M. Lui, M. Zampieri, T. Baldwin, and K. Lindén, "Automatic language identification in texts: A survey," *Journal of Artificial Intelligence Research*, vol. 65, pp. 675–782, 2019.

[9] M. D. Rau, "Language identification by statistical analysis," NAVAL POSTGRADUATE SCHOOL MONTEREY CA, Tech. Rep., 1974.

[10] C.-C. Ng and A. Selamat, "Improved letter weighting feature selection on arabic script language identification," in *2009 First Asian Conference on Intelligent Information and Database Systems*. IEEE, 2009, pp. 150–154.

[11] T. Kerwin, "Classification of natural language based on character frequency," *Ohio Supercomputer Center*, 2006.

[12] B. Ranaivo-Malançon, "Automatic identification of close languages-case study: Malay and indonesian," *ECTI Transactions on Computer and Information Technology (ECTI-CIT)*, vol. 2, no. 2, pp. 126–134, 2006.

[13] P. Henrich, "Language identification for the automatic grapheme-to-phoneme conversion of

foreign words in a german text-to-speech system," *Proceedings from EUROSPEECH-89: 220*, vol. 223, 1989.

[14] K. R. Beesley, "Language identifier: A computer program for automatic natural-language identification of on-line text," in *Proceedings of the 29th annual conference of the American Translators Association*, vol. 47, 1988, p. 54.

[15] T. Dunning, *Statistical identification of language*. Computing Research Laboratory, New Mexico State University Las Cruces, 1994.

[16] W. B. Cavnar, J. M. Trenkle *et al.*, "N-gram-based text categorization," in *Proceedings of SDAIR-94, 3rd annual symposium on document analysis and information retrieval*, vol. 161175. Las Vegas, NV, 1994.

[17] A. Bhargava and G. Kondrak, "Language identification of names with svms," in *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, 2010, pp. 693–696.

[18] A. Kulmizev, B. Blankers, J. Bjerva, M. Nissim, G. van Noord, B. Plank, and M. Wieling, "The power of character n-grams in native language identification," in *Proceedings of the 12th workshop on innovative use of NLP for building educational applications*, 2017, pp. 382–389.

[19] M. Al-Badrashiny and M. Diab, "Lili: A simple language independent approach for language identification," in *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, 2016, pp. 1211–1219.

[20] M. Wechsler, P. Sheridan, and P. Schäuble, "Multi-language text indexing for internet retrieval," in *Computer-Assisted Information Searching on Internet*, 1997, pp. 217–232.

[21] C. Souter *et al.*, "Natural language identification using corpus-based models," *HERMES-Journal of Language and Communication in Business*, no. 13, pp. 183–203, 1994.

[22] G. Grefenstette, "Comparing two language identification schemes," in *Proceedings of JADT*, vol. 95, 1995.

[23] A. K. Singh, "Study of some distance measures for language and encoding identification," in *Proceedings of the Workshop on Linguistic Distances*, 2006, pp. 63–72.

[24] G. Laboreiro, M. Bošnjak, L. Sarmento, E. M. Rodrigues, and E. Oliveira, "Determining language variant in microblog messages," in *Proceedings of the 28th Annual ACM Symposium on Applied Computing*, 2013, pp. 902–907.

[25] P. McNamee, "Language identification: a solved problem suitable for undergraduate instruction," *Journal of computing sciences in colleges*, vol. 20, no. 3, pp. 94–101, 2005.

[26] S. Konstantopoulos, "What's in a name?" vol. 408, 09 2007.

[27] T. Baldwin and M. Lui, "Language identification: The long and the short of the matter," in *Human language technologies: The 2010 annual conference of the North American Chapter of the Association for Computational Linguistics*, 2010, pp. 229–237.

[28] M. Thoma, "WiLI-2018 - Wikipedia Language Identification database," Jan. 2018. [Online]. Available: https://doi.org/10.5281/zenodo.841984

[29] N. Shuyo, "Language detection library for java," 2010. [Online]. Available: http://code.google.com/p/language-detection/

[30] A. F. Llitjos and A. W. Black, "Knowledge of language origin improves pronunciation accuracy of proper names," in *Seventh European Conference on Speech Communication and Technology*, 2001.

[31] Y. Chen, J. You, M. Chu, Y. Zhao, and J. Wang, "Identifying language origin of person names with n-grams of different units," in *2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings*, vol. 1. IEEE, 2006, pp. I–I.

[32] H. Li, K. C. Sim, J.-S. Kuo, and M. Dong, "Semantic transliteration of personal names," in *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, 2007, pp. 120–127.

[33] S. Nobesawa and I. Tahara, "Language identification for person names based on statistical information," in *Proceedings of the 19th Pacific Asia Conference on Language, Information and Computation*, 2005, pp. 289–296.

[34] C. J. Tang Lin, Guo Chonghui, "Review of chinese word segmentation studies," *Data Analysis and Knowledge Discovery*, vol. 4, no. 2/3, p. 1, 2020. [Online]. Available: https://manu44.magtech.com.cn/Jwk_infotech_wk3/EN/abstract/article_4773.shtml

[35] G. Mao, "Study on chinese word segmentation," *Advances in Higher Education*, vol. 3, p. 1, 11 2019.

[36] B. An and C. Long, "Ancient tibetan word segmentation based on deep learning," in *2021 International Conference on Asian Language Processing (IALP)*, 2021, pp. 292–297.

[37] S. Srinivasan, S. Bhattacharya, and R. Chakraborty, "Segmenting web-domains and hashtags using length specific models," in *Proceedings of the 21st ACM international conference on Information and knowledge management*, 2012, pp. 1113–1122.

[38] P. Norvig, "Natural language corpus data," 2009.

[39] Transfermarkt. [Online]. Available: https://www.transfermarkt.com/

[40] dcaribou, "transfermarkt-datasets," 2023. [Online]. Available: https://github.com/dcaribou/transfermarkt-datasets

[41] P. Remy, "Name dataset," https://github.com/philipperemy/name-dataset, 2021.

[42] D. Vrandečić, "Wikidata: A new platform for collaborative data collection," in *Proceedings of the 21st international conference on world wide web*, 2012, pp. 1063–1064.

[43] V. I. Levenshtein *et al.*, "Binary codes capable of correcting deletions, insertions, and reversals," in *Soviet physics doklady*, vol. 10, no. 8. Soviet Union, 1966, pp. 707–710.

[44] L. Mosley, "A balanced approach to the multi-class imbalance problem," 2013.