Automatic Annotation of Unstructured Fields in Medical Databases

Margarida Andreia Rosa Correia
margarida.correia@tecnico.ulisboa.pt

Instituto Superior Técnico, Lisboa, Portugal

October 2018

Abstract

The increased use of systems based on Electronic Health Records caused an enormous increment of information available electronically, which can be processed by Data Mining techniques, leading to relevant findings. The expected result was that this information becomes easy to access, analyze and share. However, the text present in the clinical notes is written in natural language, and is, thus, unstructured, and difficult to automatically process. These clinical notes might contain pertinent data for the health of the patient. In this thesis, with the help of Natural Language Processing and Information Extraction techniques, we present a system that, given a clinical note, extracts relevant named entities from it, such as names of diseases, symptoms, treatments, diagnosis and drugs, generating structured information from unstructured free text. In addition, in order to avoid privacy issues and considering that these clinical notes might contain references to names of patients, doctors or another health professionals, we also present an anonymization step. Finally, we add a module that automatically corrects typos in medical notes. Final results show that the system, in general, is able to recognize and interpret medical entities.
Keywords: Electronic Health Record, Information Extraction, Natural Language Processing, unstructured data, structured data.

1. Introduction

On a daily basis, in hospitals, a large number of medical reports is written, and much of the information they contain is in a textual format [6, 8]. Electronic Health Records (EHRs) combine information from diverse sources like notes taken in consultations between doctors and patients and radiologic or laboratory results. The record, thus, becomes an electronic merged collection of such data, organized in chronological order.

The purpose of creating EHRs is to gather essential information for each citizen to "*improve the accuracy, efficiency, quality of health care and data recorded in a health record*" [7]. To achieve this goal is necessary to register the patients, collect the data and share this information among the health entities, ensuring the confidentiality of the people involved.

However, there are still fields that are not structured (free text clinical narratives), which are difficult to analyze and most often contain valuable information for the patient's health status. These fields are shared rarely because they also include confidential patient information, such as name, address and citizen's card number. This is commonly the case with clinical notes because of remains easier to the health professionals to express there the symptoms and patients complains, as well as, to document clinical events [14] in a field dedicated to unstructured free text.

With the help of **Anonymization**, **Natural Language Processing (NLP)** (which "*is a subfield of computer science concerned with intelligent processing of human language*" [4]) and **Information Extraction (IE)** ("*which refers to the automatic extraction of concepts, entities, and events, as well as their relations and associated attributes from free text*" [14]) areas, it is possible to extract data from unstructured free text, anonymize them (ensuring that there is no loss of information, only omission) and structure them into their relevant entities, as a way to acquire structured knowledge from the unstructured clinical text. We assume that structuring a text consists of creating a frame, that is a list of attribute-value pairs, regarding relevant information extracted from text. The attributes of the frame will be the entities and the value will be the value of these entities. As entities we can have, for instance: symptoms, drugs, treatments and diseases names. For example, if we have the data "*O paciente tem febre e dor de cabeça quando acorda. Está a tomar Brufen.*" (The patient has flu and headache when waking up. Is taking Brufen.), we want to obtain as frame the Example 1.

Sintoma: febre
Sintoma: dor de cabeça
Fármaco: Brufen

Example 1: Caption

Having the text structured, besides making it easier to analyze, summarize and share, makes it easier to extract knowledge of all this data, relating them to each other using inference techniques.

1.1. Thesis Proposal and Contributions

For this work, we proposed to analyze EHRs about rheumatologic patients. We obtained the information from *"Sociedade Portuguesa de Reumatologia (SPR)"*, which gave us data from a database named *"Registo Nacional de Doentes Reumáticos"* (Reuma.pt)[1].

Rheumatism is an age-related degenerative disease, so its frequency has been increasing with the increase of the average life expectancy of populations. This disease affects the quality of life of the society because it involves bones, muscles and joints.

The challenge of this work is to **develop automatic annotation techniques in clinical notes using the methods from the NLP and IE areas**. These clinical texts contain relevant information about the health of the patient and might help in clinical decision support. There is a bunch of work into applying Data Mining techniques on texts relatively to health. However, it is easier to infer knowledge with **structured text**. The focus of our work is to struct those texts and, with our results, will be possible to use them as data to future work on Data Mining.

To obtain our final results, we need to:

- Pre-Process the texts due to the several uses of acronyms and spelling errors;
- Anonymize the texts to did not compromise the privacy of the people involved;
- Structure the texts through the relevant entities that constitute them. It will be reached using NLP and, more precisely, using IE to extract relevant information of data.

1.2. Document Structure

The organization of the contents of this document is as follows: Section 2 introduces related work of three important tasks of this work: Anonymization, Pre-Processing tasks and IE. We present systems that solve some of the problems that we found in our data, as well as medical datasets.

[1] http://reuma.pt/

Section 3 specifies our methodology and solution, including a description of the tool that we will use. In Section 4, we describe the five significant steps until we reach our goal: Dealing with Acronyms, Anonymization, NER, Spell Checking and Structuring the Clinical Notes. We explain the challenges and our approach. In Section 5, we exhibit our experimental evaluation, applying the standard evaluation metrics to the corpora chosen. Finally, Section 6 summarizes the main contributions of this dissertation and presents ideas for future work as, also, our work limitations.

2. Related Work

The architecture of an IE process normally has two modules: Pre-Processing and the IE task [1]. In this work, in addition, to follow this architecture, we still need to implement, first of pre-processing, the anonymization task because we have in hands confidential information that we can not compromise. In this section, we present systems, competitions, datasets and papers for these three principal tasks, as also a final section that explains how these systems influenced our solution.

2.1. Anonymization

Medical records are delicate and, before undergoing transformations or extractions, must be anonymized in a way that does not compromise patient's privacy and data's integrity. When we consider NER tools, data anonymization consists of finding named entities that we identify as pertinent to our domain and applying some techniques to change this information in another expression. These systems based on NER can be either rule-based (dictionaries or pattern-matching), model-based (Machine Learning (ML) models) or hybrid (a combination of these two techniques) [11].

2.1.1. Anonymization with STRING chain

The STRING chain [10] is a hybrid system (later explained in this documentt) and is divided into four different steps: Pre-Processing, NER, Coreference Resolution (CRR) and Anonymization.

In the **Pre-Processing** step, is used the STRING to normalize texts, separating it into sentences and tokens. In the **NER's** module, the STRING is also used to obtain the list of named entities. This system is able to anonymize names, localizations and organizations. Next, **CRR** is used to verify if two different named entities refer to the same object in a way to not be replaced by different terms. Considering titles and abbreviations as *"Maria Silva"* and *"Sra. Silva"*. We know that these two different writing ways, in the same context, refer to the same person, and the system knows that too.

In the last module (**Anonymization**), we can choose between suppression, tagging, random substitution and generalization methods. **Suppression** consists in the omission of the NE, using a symbol or character that replaces the original text, for instance, we could change *"Maria"* by *"XXXXX"*. This is the most common method used by anonymization systems [9]. **Tagging** is similar to Suppression, but instead of changing the original text by a symbol or character, it substitutes it for a label that indicates its class. Continuing with the Maria example, the anonymization returning from tagging might result in *"[**PersonName01**]"*. In the **Random Substitution** method, the system replaces a Named Entity (NE) with another NE, respecting the same genre. The last method, **Generalization**, by default, can not be applied to entities that represent person names, but consists of "*replacing an entity by another that mentions an item of the same type but more general. e.g. University of Lisbon becomes University*" [9].

2.2. Pre-Processing

The tasks of pre-processing must be done carefully to avoid losing important information and, if successfully done, the IE task will become easier because it significantly reduces the search space and time, facilitating the data analysis in this next step. In this section, we present two Pre-Processing methods helpful for our work.

2.2.1. Spell Checking

The health professionals write the reports with misspelled words, requiring some type of correction either manual (which will be time-consuming), automatic or semi-automatic.

There is a tool that adopts a semi-automatic detection and correction of misspelled words [2]. The tool has a corpus of common words in the domain – **Known Words List (KWL)** – and a corpus of words that appear in the report ordered alphabetically – **Corpus Words List (CWL)**. The tool finds similarity between the words of KWL and CWL, ordering these words by similarity in a **filtered Corpus Words List (f-CWL)** to filter small typos. Next, the authors need to separate f-CWL into **High Frequency List (HFL)** and **Low Frequency List (LFL)** based on thresholds. In HFL they assume that the words that occur many times do not contain many errors because they filter these words in the previous step. They will use this list as the KWL in the correction final step. In the LFL they assume that the words that occur a small number of times have errors or are abbreviations or technical terms. They will correct this list with HFL. The unique disadvantage of this tool is the work to create the KWL and the CWL.

2.2.2. Word Sense Disambiguation

In the medical domain it is frequent the use of acronyms and abbreviations. The disambiguation of these terms is a sub-task of WSD [12]. For instance, *MCF* can refer to "*metacarpofalangeana*" or to "*Moto Club Faro*" (as we are in a medical domain, the most probable meaning of this acronym will be the first one). This example situation is easy to solve due to the context in which we are and because we suggest have two options of extension of the acronym. But it is very common in the medical context that an acronym can come to have several meanings, all of them within the context of medicine [8] that demonstrated that 81% of acronyms found in MEDLINE [2] abstracts are ambiguous and have on average 16 senses, which makes difficult the work of disambiguation. A solution to that is considering a "*global context*" [12], that is, within the medical context, see in what context the acronym arises. For example, if we are reading a medical report that is talking about a heart problem, it is more normal for "*RA*" to mean "*right atrial*" than "*rheumatoid arthritis*". This ambiguity requires a rule-based system to choose the correct acronym's extension and also a database of acronyms with all their true extensions.

2.3. Information Extraction

After processing the texts, we are ready to apply IE techniques to our data. We present two systems to do the IE task and the datasets used.

2.3.1. Natural Language Text Processor for Clinical Radiology

This tool of NLP was made to identify clinical information in radiologic reports, structuring that [5]. It is composed for three processing phases: **Parser**, **Phrase Regularization** and **Encoder**.

The **Parser** is used to determine the structure of the text, following the rules of a semantic grammar. This phase is complicated because there is no "*single nomenclature*". In this process, all variations of the same term have to give rise to the same results, but "*not all variations are reduced to one form by this stage of processing, and the structured forms do not yet correspond to unique controlled vocabulary concepts*" [5] thus, in the **Phrase Regularization** phase a Mapping composed of multiword is used to reduce these linguistic variations to a default form defined for the words.

In the last phase, **Encoding**, a list of synonyms, with terms taken from the Medical Entities Dictionary (MED)[3], was used to map all the variations of a concept into a single general concept.

[2] https://www.nlm.nih.gov/bsd/medline.html
[3] http://med.dmi.columbia.edu/

2.3.2. cTAKES

The Clinical Text Analysis and Knowledge Extraction System (cTAKES) is an NLP system, developed on 2013 by Mayo Clinic, to extract events and clinical concepts from the text. This system is based on the Unstructured Information Management Architecture (UIMA) framework[4] and works with EHRs unstructured clinical texts.

It combines rule-based and machine learning techniques (hybrid system) and is composed of six modules: **Sentence Detection**, **Tokenization**, **Normalization**, **Part-of-Speech Tagging**, **Shallow Parser** and a **NER annotator** [13]. First, the system detects the sentence (Sentence Detection), next it divides it into its constituent tokens (Tokenization). After that it normalizes all the tokens (Normalization), that is, transforms every word into its basic form (lemma). Next, it identifies the constituent parts of sentences as nouns, verbs and determinants (Part-of-Speech Tagging), and, after that, links them into nodes related syntactically as noun groups and verb groups (Shallow Parser). Finally, cTAKES recognizes the entities and find if they are negated or not (NER annotator).

To detect sentences, the system finds where the punctuation that implies the end of a sentence (full stop mark, question or exclamation marks) is. Relatively to the tokenizer it has two components: one which splits the sentence into spaces and punctuation and another which merges the tokens that do not make sense to be split (as in dates and hours). To normalize the words, the authors resorted to a dictionary of the SPECIALIST lexical tools[5], which gives the lemma of the words. Next, the Part-of-Speech Tagging and the Shallow Parser are "*wrappers around OpenNLP's modules for these tasks*" [13]. Finally, the NER module uses dictionaries with concepts from the SNOMED CT[6] and RxNORM[7].

The entities to be recognized are: disorders/diseases, symptoms/signs, procedures, anatomy and drugs. In relation to the negation attributes, cTAKES uses the NegEx algorithm [3], which finds if there is any negative term near to the entities that could negate them.

2.4. Overview

In this section we explain why we choose these before mentioned tools and how did they help us. Relatively to the anonymization step, we concluded that we will use the **STRING chain** because it is prepared to texts written in the Portuguese language and we have access to it.

Regarding the Pre-Processing tasks:

- In the **Spell Checking**, we present that the idea of correcting words is an important task;
- With the task of **WSD**, we inferred that is important to disambiguate the acronyms and abbreviates. We also concluded that it needs to be done basing on the context and, if they are ambiguous, using rules.

Although we have demonstrated two tools that apply IE in the clinical context, we are conscious that one of the biggest difficulties of this work is due to the text is written in the Portuguese language and the current state-of-art is bigger for English texts. However, we retrieved some ideas of these tools:

- The **Natural Language Text Processor for Clinical Radiology** tool shows that using a database that is formed by various medical terms it is possible to contour the problem of not having a *"single nomenclature"*;
- The **cTAKES** tool applies rules and has modules related to the ones that we want for our work. It has a way of acting similar to that of the STRING chain, helping us to understand what types of entities are essential to structure, as also to create a special attention for the case of the negation attributes.

3. Methodology and Proposed Solution

In this section we will describe the STRING chain tool and explain how we will deal with the problems and the challenges that we have identified.

3.1. The STRING chain

This is an in-house tool of *Instituto de Engenharia de Sistemas e Computadores - Investigação e Desenvolvimento (INESC-ID)*, the local where I developed my thesis. STRING is an hybrid, statistical and rule-based NLP chain for the Portuguese language [10]. It adopts an architecture based on modules, as we can see in Figure 1 – taken from the STRING site[8].

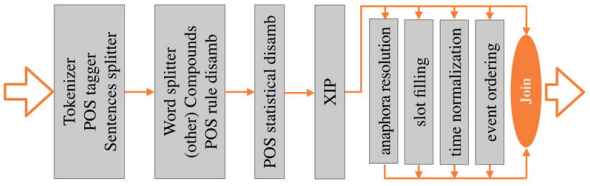

Figure 1: STRING chain architecture

The first module receives the input text and separates it into their respective fragments (tokens).

[4] https://uima.apache.org/
[5] https://lexsrv3.nlm.nih.gov/Specialist/Home/index.html
[6] https://www.snomed.org/
[7] https://www.nlm.nih.gov/research/umls/rxnorm/
[8] https://string.l2f.inesc-id.pt

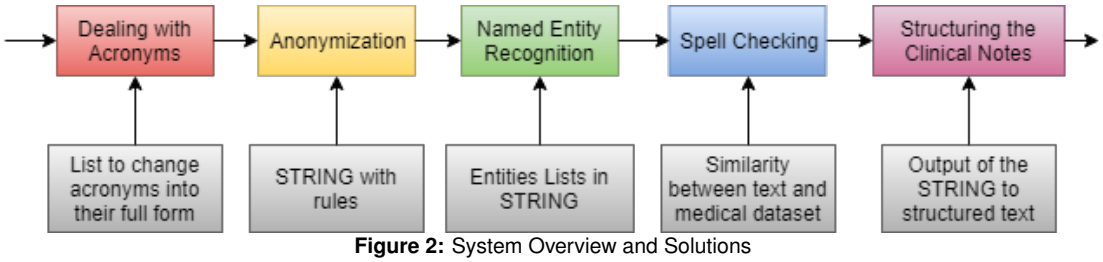

Figure 2: System Overview and Solutions

These tokens are the input from the second module that applies the LexMan[9] morphological tagger to assign to every token its morphological category.

The third module receives as input the segments grouped into sentences and consists of applying the RuDriCo2[10] to makes disambiguation and segmentation rules. The disambiguation is used when a word can be from two or more morphological types. The disambiguator needs to choose the correct category, considering the surrounding text. The segmentation rules can be subdivided into expansion and contraction rules. Expansion rules consist of converting a segment into two or more fragments. Contraction rules consist of transforming two or more segments into only one.

The last module applies the XIP[11] parser for the syntactic analysis. This parser allows us to locally apply our disambiguation rules and also add lexical, syntactic and semantic information.

Hereupon, we can validate that STRING was very useful for this work because it performs NER and anonymization tasks and enables us to add more semantic information, helping us to process our medical text.

3.2. System Overview

We divide our solution in five steps, shown in Figure 2: Dealing with Acronyms, Anonymization, NER, Spell Checking and Structuring the Clinical Notes.

Primarily, we do a list of pairs acronym-extensive full form in a way to substitute the acronyms by their extensive full form in the original text. Secondly, we solve the anonymization step using and applying rules on the STRING chain. After that, we do lists of terms that compose all the entities mentioned before and insert it on the STRING chain. It helps us to easily recognize the entities. In the fourth step, to correct the writing errors in the clinical notes, we adopt an idea of an article using a measure of similarity that compares our text with a medical dataset [2]. Lastly, after having our problems solved and the clinical notes processed, we will use the output of STRING to structure these texts, obtaining the desired frames.

[9] https://string.l2f.inesc-id.pt/w/index.php/LexMan
[10] https://string.l2f.inesc-id.pt/w/index.php/RuDriCo2
[11] https://string.l2f.inesc-id.pt/w/index.php/XIP

4. Processing the Clinical Notes

In this chapter, we will explain in detail the five steps that we take until we reach our final goal, that is, structure the text of the clinical notes.

4.1. Dealing with Acronyms

In this first step we made a list of acronyms and their respective full form. There are acronyms already inserted on the STRING that will create ambiguity with ours, although they have different meanings. For instance, *"AV"* is inserted on the STRING, representing an abbreviation to *"avenida"* (avenue) and, in a clinical context, we want that it represents *"átrio ventricular"* (ventricular atrium).

The best solution that we found to overcome this problem was to make this list of medical acronyms and replacing them in the text with their full form, before moving on to the next step. The final list is composed of about 450 acronyms.

In addition to the problem that medical acronyms create ambiguity with "normal" acronyms on the STRING, medical acronyms have ambiguities between them too, as *"AU"* that can represent *"altura uterina"* (uterine height) or *"ácido úrico"* (uric acid). We need to know in which textual situation we use one or another.

4.2. Anonymization

To avoid compromising the privacy of the people involved, we need to anonymize this data. However, we must take into account that: there can be no proper names, but we must ensure that there is no loss of information.

We choose to randomize the name of the person, because, in the next steps, the system would have to know that this word was a person's name.

As input, we pass the clinical note without acronyms and, as output, we obtain the text without any reference to someone. Various problems of bad anonymizations were detected due to:

1. **Medical writing style:** Several uses of acronyms and abbreviations that the system confuses with people names;

2. **Lack of punctuation:** A new sentence starts with a capital word, but the previous one does not end with a full stop. The system assumes the capitalized word as a person's name;

3. **Hospital and drugs names:** The system often considers as person's names, hospital names like *"Santa Maria"* and drug names as *"Humira"*;

4. **Portuguese specificities:** In some cases, the system confuses a Portuguese verb with a person name. An example is *"Marco consulta"* (Schedule a consultation). It appears several times and the system assumes that *"Marco"* is a male name (which is, indeed).

4.3. Named Entity Recognition

The solution that we found to recognize the entities that we consider relevant is through lists of terms. These lists were done based on external information (present on the web) and internal information (present on the database of *Reuma.pt*).

We generated lists of: **rheumatic diseases** (19 entries) with terms as *"Gota"* and *"Síndrome de Sjögren"*, **clinical problems** (1803 entries) with terms as *"abcesso abdominal"* (abdominal abscess) and *"pancreatite"* (pancreatitis), **human body** (174 entries) with terms related to bones and articulations as *"joelho"* (knee) and *"metacarpofalangeana"* (metacarpophalangeal) and **diagnose exams and tests** (104 entries) with terms as *"eletrocardiograma"* (electrocardiogram) and *"mamografia"* (mammography).

After, we gathered lists of: **treatments** (143 entries) with terms as *"cateterismo"* (catheterization) and *"laparotomia"* (laparotomy) and **hospitals** (83 entries) with terms as *"Hospital da Luz"* and *"Hospital de Braga"*. These two lists were made based on both internal and external information.

The lists based on external information were created based on *Wikipedia*[12], on the general web and sites like the *SPR*[13] and *Atlas da Saúde – Listas de Medicamentos do Infarmed*[14] (to retrieve the medicines names). The gathered lists based on only external information are: **drugs** (1314 entries) with unigrams as *"Clavamox"* and *"Ibuprofeno"*, **drugs brands** (160 entries) with unigrams as *"Omezolan"* and *"Ratiopharm"*, **bacterias** (178 entries) with unigrams as *"Enterococcus"* and *"Yersinia"*, **active substances on medicines** (717 entries) as *"Naloxona"* and *"Tramadol"* and **hormones** (83 entries) as *"Prostaciclina"*.

Lastly, we end this point with an exhaustive list of medical acronyms and abbreviations present in these texts (492 entries) with their respective extensive full form. This list contains pairs as *"ADM – Amplitude de Movimento"* (Range of Motion) and *"SZP – Salazopirina"*.

[12]https://www.wikipedia.org/

[13]http://www.spreumatologia.pt/

[14]http://www.atlasdasaude.pt/lista-de-medicamentos-infarmed

After a preliminary evaluation of these lists on the STRING chain, we concluded that we need to add more rules and dependencies in specific situations. For instance, as we are doing text analysis and not text generation, we assume that the doctors do not write "abnormal" things as *"nariz direito"* (right nose). So, we wrote a rule that says that if we found any term that has the tag *"SEM-corpoHumano"* and it has *"direito"* (right) forward, we continue in a presence of a human body part. We do the same rule but with the *"esquerdo"* (left) term too, considering that any part of the human body could have a right and a left side. We created a total of 75 rules.

4.4. Spell Checking

In the medical texts, typographical errors (as substitution of one letter for another, transposition and omission) are usual. To correct these typos, we follow the idea of the previously mentioned article of Spell Checking. They created a semi-automatic error detection based on a bag-of-words model. For that, firstly, we need to do a script that returns all the different existent unigrams that constitute the reports, creating the CWL. To create the KWL we download a corpus of Portuguese text from *Linguateca*[15]. Next, we run the script that returns the uniques unigrams from this corpus. We add to the unigrams list the terms from our drugs, brand drugs and active substances lists because these terms are very usual on the texts, and they are not present on the *Linguateca* corpus.

The final list of KWL has a total of 123M terms and the final list of CWL has a total of 56K terms.

We needed to know which unigrams are well written and which are not. For that, we compared all the unigrams of the CWL with all the unigrams of the KWL. If they are equal, it means that they are well written. After this comparison, we obtained a list of 16036 correct unigrams. However, it does not imply that all the others are wrongly written: they could only be out of our KWL list.

Then, we made a count of all the unigrams of the remaining CWL list (39793 unigrams) – f-CWL of the Figure 1.

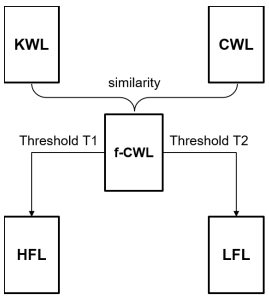

Figure 1: Schema of the writing errors correction

[15]https://www.linguateca.pt/

The terms that usually appear (with the highest score) only have accentuation problems or did not appear in our dataset. Regarding again the same article, they made the following assumptions:

- The unigrams that appear 5 or fewer times are wrongly written (LFL), so we apply the Jaro distance between them and our dataset;
- The unigrams that arise more than 5 times may not have been caught by the algorithm because they do not appear on the Medical Dataset. These unigrams are on the HFL.

The unigrams that appear on the LFL will be replaced by the unigrams that have higher similarity.

We conclude that the 73221 original clinical notes contain 55829 unigrams of which 40510 are possibly misspelled. Firstly, we applied this algorithm to the unigrams that appear less than 5 times (including). In this situation, there are a total of 35028 different unigrams that are possibly misspelled. This gave us a total of 25930 best suggestions. Of this total of 25930 best suggestions, we obtained 21240 right substitutions (the right substitution for a word is the maximum similarity that we found between the best suggestions for that word).

We only executed this algorithm to the 15000 anonymized clinical notes. These data contain 24169 different unigrams. The 73221 original clinical notes, contain 21240 right substitutions. In common, the 15000 data and the 21240 right substitutions have 5376 unigrams, that is, there are 5376 different terms that are misspelled in these 15000 clinical notes.

For these 5376 unigrams, the ones that occur less or equal than 5 times originated a total of 5744 substitutions in the anonymized data. For the remaining unigrams (the ones that occur more than 5 times), we found that for the 4621 best suggestions originated by them, 3522 are the right substitution, according to the algorithm. For the 3522 right substitutions, we substitute in the original text (15000 data anonymized) 39559 times. There are terms that do not receive any best suggestion because:

- None of the words of the dataset have a similarity with them, higher than the threshold. It can occur because the well-written word may not exist in our Portuguese dataset;
- The fact that some substitutions are not correct may also be due to the same case. For example, "*invaginação*" is a well-written word but, as the dataset does not contain this, the algorithm suggests the most similar word to that, that is "*inalação*".
- The Jaro measure assigns more weight by making substitutions for accented letters than removing letters. For instance, *"['comunicacao', 'comunica', 0.9090909090909092]"*. The right substitution will be "*comunicação*" (comunication), but as taking 3 characters is heavier than adding 2 accented letters, the distance between these words would be smaller, which gives a greater similarity.

4.5. Structuring the Clinical Notes

We present how we converted the free text of the clinical notes into a structured text, by using the output of the STRING chain. It has several types of outputs as syntactic trees, syntactic nodes information, showing the dependencies between the words and XML. To our work and to obtain our final output of the structured text, the simplest and best to use from all the outputs that the STRING provides us is the XML output. Unfortunately, there was a bug on the STRING when we process some text that uses the rules made by us. This text did not come with the associated tags that we define, despite the rules triggering in these cases. Thus, we choose to parse the output in another format, the one that shows the dependencies between the words.

After we have developed an initial version of our code to process the output of the dependencies and relations between the terms, we test, passing as input a small number of clinical notes and we conclude that we already have some terms to introduce on the STRING chain, that we do not notice before, as diagnose terms like *"colonoscopia"* (colonoscopy) and *"endoscopia"* (endoscopy), drugs names as *"Fucithalmic"* and *"Pneumo 23"*, among others.

Furthermore, we still need to improve our grammar, adding more rules, to get the desired output without losing relevant information for the health of the patient. In the total were added 244 new rules.

5. Experimental Results

We used for evaluation a portion of text never seen before by us, from the data of *Reuma.pt*. **300 clinical notes** were randomly choosen of the remaining non-annotated data (58221 clinical notes). We prepared this text as a gold collection, that is, we annotate, for each of the steps, the desired output.

Firstly, we replace the acronyms with their expansion in full form. We then use this changed text to manually annotate the names of people that the system is supposed to find and apply the annonymization. We made a total of **28 annotations of people names**, and we got this way the gold collection for the anonymization step.

Next, we moved to the spell checking. We applied our algorithm to the 2817 different unigrams that meet the size requirements. Manually, we

identified **339 substitutions to misspelled words** as *"aidna"* which the well-written word is *"ainda"*.

We end this preparation of the evaluation corpora making a gold collection to the last step of this work – Structuring the Clinical Notes. As referred, what we want in the final is to obtain the tags associated with the medical terms. When structuring these evaluation corpora, we obtained the following number of tags, shown on Table 1.

Entities	Number of Tags
Fármacos	344
Doenças Reumatológicas	43
Diagnóstico	635
Bactéria	6
Hormona	2
Hospital	68
Sintoma/Problema Clínico	867
Tratamento	89
Substância Ativa	221
Marca de Fármaco	0

Table 1: Number of entities on the evaluation corpora

5.1. Results

We compared each of the outputs defined by us in the gold collections previously mentioned, with theoutput returned by the system.

5.1.1. Anonymization Evaluation

As already mentioned, we manually annotate the **28 person names** that we found in this corpora. Running this same text in the STRING chain, randomizing the names that this tool is encountering, we obtained a **total of 91 anonymized names**. Using this information and applying the previously explained formulas, we obtained the next results:

$$Precision = \frac{28}{(28+63)} = 0.31$$

$$Recall = \frac{28}{(28+0)} = 1$$

$$F - measure = 2 * \frac{0.31 * 1}{(0.31+1)} = 0.47$$

The erroneous anonymizations were due to:

- Misspelled words that the system confuses with a personal name. **Example:** The doctor wrotes *"Urian II"* instead of *"Urina II"*. So the system did not captured the NE *"Urina"* and considers *"Urian"* as a proper noun because it stars with a capitalized letter;
- Medical tests and experiences with personal names. **Example:** There is a test diagnose whose name is *"ELISA"*, which it is also a Portuguese name and the system assumes the exam name as being a person;
- Acronyms that were not replaced for their expansive full form because neither we nor the doctors know the real expansive full form from them. **Example:** *"DDCPC"*;
- According with the information provided to the system, some acronyms were missing. **Example:** *"UII"* that means *"Urina II"*;
- Lack of punctuation that generates an error in the system. **Example:** *"Azitromicina Imonugenicidade Junho 2016"*, the system gives as output *"[**Matilde**] Junho 2016"*, because it does not recognize the name *"Imonugenicidade"*. If we run only the term *"Azitromicina"*, the system will know that it is a drug name, but, as we insert it on the STRING without punctuation, the system joins all as one sentence and it gives a bad result.

5.1.2. Spell Checking Evaluation

As we previously wrote, we only applied our algorithm to the **2817 different unigrams** whose size is between 3 and 14 characters (including). After comparing these unigrams with all the unigrams of the Portuguese dataset created by us, we obtained as a result that 2200 unigrams are possibly well-written (appear in the dataset), so, only 617 are misspelled. To these 617 unigrams, a list of drugs was compared, that gave us as a result that, after all, is only **596 the misspelled unigrams**.

When running the algorithm with a threshold of similarity bigger than 0.8 (that is, 80%), 953 changes of terms were suggested. Of these suggested changes we only choose the one with bigger similarity for each word (best suggestion), obtained a total of 238 suggestions of change.
The remaining 358 terms of the misspelled 596 not have any suggestion possibly due to:

- The corrected words do not appear in our Portuguese dataset. **Example:** *"artralgias"* and *"gonalgias"* are well-written terms, but do not occur as suggestions because they are not present in the Portuguese dataset done by us.
- Some terms are in English and we do not have any Portuguese terms with a similarity bigger than 80% with English terms. **Example:** *"ankylosing"* and *"spondylitis"* are English terms presents on the clinical notes.

As we already referred, we identified manually the **339 misspelled terms**, indicating the possible well-written word that should arise. Our algorithm

Entities	Number of Entities on the Gold-Collection	Number of Entities given by the STRING Chain	Precision (%)	Recall (%)
Fármacos	344	233	93.1	63.1
Doenças Reumatológicas	43	29	100	67.4
Diagnóstico	635	556	88.9	77.8
Bactéria	6	3	100	100
Hormona	2	1	100	100
Hospital	68	66	97	94.1
Sintoma/Problema Clínico	867	620	98	70
Tratamento	89	75	100	84.3
Substância Ativa	221	155	98.1	68.8
Marca de Fármaco	0	0	100	100

Table 2: Precision and Recall metrics applied to the structure

run with the same corpora and gives as output 238 misspelled terms, indicating, also, the possible well-written word that should arise in that situation. Of these 238 suggested words by the system, only **105 are in agreement with our manual suggestions**. We list some situations that originate this number:

- The system still correctly identified 44 misspelled terms, but do not suggests well the well-written word that should appear (gives us a bad suggestion of correction, although it originates the bigger similarity with the misspelled word);
- Some suggestions provided by the system are well-suggested and we do not notice that these terms are misspelled, so we do not put them in our manual suggestions;
- In these 238 suggestions provided by the system, 120 are not totally correct but are cases as, for instance, the misspelled word be in the plural and the suggestion of correction is good, but appear in the singular.

Using these data and applying the standard formulas, we obtained the next results:

$$Precision = \frac{105}{(105 + 103)} = 0.44$$

Recall in this case is 1 because the system never classifies a term as correct without this being, that is, it always ranks correctly all the correct terms.

5.1.3. Entities and Structure Evaluation

To do the Structure Final Evaluation, we recurred to the Table 1 presented above, that shows us what is the supposed number of each entity that is identified by the system, basing on our gold-collection.

After running on the STRING chain our 300 clinical notes for evaluation and parsing the result to a structured format, was possible to compare the gold-collection with the STRING result, in order to do the following Table 2, using an auxiliary Table 3.

Entities	TP	FP	FN
Fármacos	217	16	127
Doenças Reumatológicas	29	0	14
Diagnóstico	494	62	141
Bactéria	3	0	3
Hormona	1	0	1
Hospital	64	2	4
Sintoma/Problema Clínico	607	13	260
Tratamento	75	0	14
Substância Ativa	152	3	69
Marca de Fármaco	0	0	0

Table 3: Auxiliar table to help in Precision and Recall results

6. Conclusions

The main contribution was to construct a pipeline with these five stages that lead us to our goal:

- **Dealing with Acronyms:** It was made a list with 492 entries of acronyms. This list allows us to substitute an acronym by their full form, hence facilitating the next stages;
- **Anonymization:** Changes were made in the STRING original code to only anonymize our sensitive information – person names;
- **Named Entity Recognition:** STRING was enriched with medical terms. In Table 4 we showed the number of entries added. It was also added 75 rules to enlarge the terms;
- **Spell Checking:** It was possible for us to correct misspelled terms, using as auxiliary some lists made by us in the previous step;
- **Structuring the Clinical Notes:** It was made more 244 rules to catch larger expressions that we want to use as value in the frame. Furthermore, we did a parser for the output of STRING chosen by us.

Entities	Number of Entries
Fármacos	1314
Doenças Reumatológicas	19
Diagnóstico	104
Bactéria	178
Hormona	83
Hospital	83
Sintoma/Problema Clínico	1803
Tratamento	143
Substância Ativa	717
Marca de Fármaco	160
Corpo Humano	174

Table 4: Number of entities that we add to the STRING

Regarding to future work, we considered:

- Apply the Name Entity Normalization task in the clinical notes. This consists of choosing a specific standard to each one of the medical terms that can be mentioned in two or more ways, doing a link between all these forms and choosing one of them to be a standard;
- Instead of structuring all the clinical notes by chronological order and randomly by patients, it can be structured by patient ID, in order to have access to a structured information from the clinical notes by the evolution of medical appointments by patients;
- With the data in a structured format, we could resort to the application of inference based techniques on this extracted data, more precisely, Data Mining.

References

[1] S. Bird, E. Klein, and E. Loper. *Natural Language Processing with Python*, volume 43. O'Reilly Media, 2009.

[2] J. P. Carvalho and S. Curto. Towards unsupervised word error correction in textual big data. *6th International Conference on Fuzzy Computation Theory and Applications, FCTA 2014, Part of the 6th International Joint Conference on Computational Intelligence, IJCCI 2014*, pages 181–186, 2014.

[3] W. W. Chapman, W. Bridewell, P. Hanbury, G. F. Cooper, and B. G. Buchanan. A simple algorithm for identifying negated findings and diseases in discharge summaries. *Journal of Biomedical Informatics*, 34(5):301–310, 2001.

[4] E. Ford, J. A. Carroll, H. E. Smith, D. Scott, and J. A. Cassell. Extracting information from the text of electronic medical records to improve case detection: A systematic review. *Journal of the American Medical Informatics Association*, 23(5):1007–1015, 2016.

[5] C. Friedman, P. O. Alderson, J. H. Austin, J. J. Cimino, and S. B. Johnson. A general natural-language text processor for clinical radiology. *Journal of the American Medical Informatics Association*, 1(2):161–174, 1994.

[6] R. Grishman. Information Extraction: Capabilities and Challenges. *International Winter School in Language and Speech Technologies*, page 377, 2012.

[7] N. A. Latha, B. R. Murthy, and U. Sunitha. Electronic Health Record. *International Journal of Engineering Research & Technology*, 1(10):1–8, 2012.

[8] H. Liu, S. B. Johnson, and C. Friedman. Automatic resolution of ambiguous terms based on machine learning and conceptual relations in the UMLS. *Journal of the American Medical Informatics Association*, 9(6):621–636, 2002.

[9] N. Mamede, J. Baptista, and F. Dias. Automated anonymization of text documents. *2016 IEEE Congress on Evolutionary Computation, CEC 2016*, pages 1287–1294, 2016.

[10] N. Mamede, J. Baptista, C. Diniz, and V. Cabarrão. STRING: An Hybrid Statistical and Rule-Based Natural Language Processing Chain for Portuguese. *PROPOR 2012 - 10th International Conference on Computational Processing of Portuguese, series Demo Session*, pages 2–4, 2012.

[11] S. Meystre, F. Friedlin, B. South, S. Shen, and M. Samore. Automatic de-identification of textual documents in the electronic health record: a review of recent research. *BMC medical research methodology*, 10(1):70, 2010.

[12] S. Pakhomov, T. Pedersen, and C. G. Chute. Abbreviation and acronym disambiguation in clinical discourse. *AMIA ... Annual Symposium proceedings / AMIA Symposium. AMIA Symposium*, 2005:589–93, 2005.

[13] G. K. Savova, J. J. Masanz, P. V. Ogren, J. Zheng, S. Sohn, K. C. Kipper-Schuler, and C. G. Chute. Mayo clinical Text Analysis and Knowledge Extraction System (cTAKES): Architecture, component evaluation and applications. *Journal of the American Medical Informatics Association*, 17(5):507–513, 2010.

[14] Y. Wang, L. Wang, M. Rastegar-Mojarad, S. Moon, F. Shen, N. Afzal, S. Liu, Y. Zeng, S. Mehrabi, S. Sohn, and H. Liu. Clinical information extraction applications: A literature review. *Journal of Biomedical Informatics*, 77(June 2017):34–49, 2018.