

Medical prescription classification: a NLP-based approach

Viincenza Carchiolo, Alessandro Longheu
Universita di Catania
Email: vincenza.carchiolo@unict.it
alessandro.longheu@dieei.unict.it

Giuseppa Reitano, Luca Zagarella
Previnet s.p.a. Treviso, Italy
Previmedical s.p.a. Treviso, Italy
Email: {giuseppa.reitano, luca.zagarella}@previnet.it

Abstract—The digitization of healthcare data has been consolidated in the last decade as a must to manage the vast amount of data generated by healthcare organizations. Carrying out this process effectively represents an enabling resource that will improve healthcare services provision, as well as on-the-edge related applications, ranging from clinical text mining to predictive modelling, survival analysis, patient similarity, genetic data analysis and many others. The application presented in this work concerns the digitization of medical prescriptions, both to provide authorization for healthcare services or to grant reimbursement for medical expenses. The proposed system first extract text from scanned medical prescription, then Natural Language Processing and machine learning techniques provide effective classification exploiting embedded terms and categories about patient/doctor personal data, symptoms, pathology, diagnosis and suggested treatments. A REST ful Web Service is introduced, together with results of prescription classification over a set of 800K+ of diagnostic statements.

I. INTRODUCTION

In recent years, there has been an amplified focus on the use of Artificial Intelligence (AI) in E-health. There are numerous examples that include AI approaches to analyze unstructured data such as photos, videos, physician notes to enable clinical decision making; or the use of intelligent interfaces to enhance patient engagement and compliance with treatment and predictive modelling to manage patient flow and hospital capacity/resource allocation.

Two main information sources play a relevant role in healthcare field, i.e. images and natural language. The use of Natural Language Processing (NLP) found several applications related to medical ICT with the increasing adoption of Electronic Health Records (EHRs); in the last decade, a lot of application have been developed in order to extract information and knowledge from electronic EHRs [1] [2]. In fact, when structured data is stored in an EHR, it is desirable to support automated systems at the point of care, and to help physicians in diagnosis. These studies endorsed most NLP applications in the medical field; for instance, those concerning the use of Twitter data and sentiment analysis to study diseases dynamics [3], or [4], where the correlation among "stress", "insomnia", and "headache" is analysed. In the field of medical application, the image processing are very useful in EHR data manipulation [5] [6], where medical images play an important role in particular to help physicians to monitor the evolution of complex pathologies [7]. In this

work a combination of image processing and NLP techniques are exploited to extract information from a scanned image of a medical prescription and analyze the semantics of the embedded information with the final goal of assessing its correctness according to the "medical request service" related to the prescription being examined. The system performs a classification in order to automatically authorize or not the medical service required within the prescription. Indeed, in Italy there exist a public medical assistance that provide free "Medical services". These though have to comply with certain parameters to be freely provided. Currently, the assessment of compliance with these parameters is manually performed by a proper operator. The proposed application aims to provide a mechanism to help the operator, or even replace his/her intervention. The proposed solution provides an user-friendly application to help the operator with a pre-analysis to isolate the few medical prescriptions that require a human operator to decide about their correctness, trying to automatize as many prescriptions as possible.

In Section II an overall description of the system with some implementation details is provided. Section III and IV describe respectively image pre-processing operations and text extraction. Section V discusses about the solution used in spelling correction and section VI describes the information classification task. Results are presented in section VII, while section VIII highlights conclusive remarks also outlining some future works.

II. SYSTEM ARCHITECTURE

In this section we illustrate the proposed system and the solutions used to achieve the goal described in the introduction. As shown in fig. 1, the system is accessible via a Web application that works according to the following steps. First, the input image is examined to establish the type and format of the medical prescription that image represents, then, text is collected and corrected to further isolate and extract all relevant strings and the information based on previously collected strings is classified. Finally, the prescription is eventually considered as valid for further approval or not, according to specific criteria based on the information and related classification; we named these two possibilities as *grantable* and *not-grantable* respectively The ASP.NET framework has been adopted to develop the whole application; in particular,

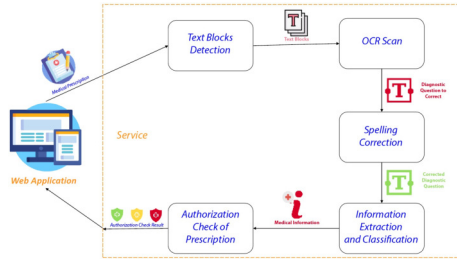


Fig. 1. Functional architecture of the proposed system

the ASP.NET Core version was considered, thanks to its support to open-source and multiplatform environments. In the paragraphs below, we discuss about each system module shown in fig. 1.

III. MEDICAL PRESCRIPTION RECOGNITION

The first module of the proposed system aims to discriminate which type of medical prescription is being processed. The *type* is defined by the Italian national medical service ("Servizio Sanitario Nazionale", simply SSN in the following), indeed it includes:

- the prescription used to provide drugs, therapies, screenings or specialist examinations at the expense (entirely or partially) of the SSN; this prescription can be filled in by physicians that either works inside SSN structures (e.g., public hospitals) or they hold an agreement with SSN (being therefore a *partner* of SSN itself)
- the prescription where any medical care as those listed above are completely at the expense of the person that prescription was written for; in such cases, the physician is not required to hold any agreement with the SSN

The former type is also known as "ricetta rossa" (*red* prescription), and it is a specific prescription whose details also depend on local (region-based) rules, whereas the latter, known also as "ricetta bianca" (*white* prescription) has a general validity on the entire national territory, therefore in the rest of paper we just focus on this last type.

The system receives as input scanned images of prescriptions that must be classified as *white* ones or discarded. To accomplish this task, a supervised machine learning approach [8] is adopted. In particular, this well-known technique exploits a training set used to build a model that enables the classifier to perform the discrimination. Since we focus on white prescriptions only, the classifier is *binary*, i.e. it just establishes whether an image actually can be considered as a white prescription or not. To assess the effectiveness of the classifier, the widely adopted 75/25 approach has been considered as the training/validation set splitting. In addition, to prevent the overfitting problem, a data augmentation [9] has been performed on the dataset; note that due to its reduced dimension, we did not consider the cross-validation technique. The classifier we developed allows to detect white prescriptions with an effectiveness of about 93-95%. After that

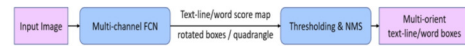


Fig. 2. EAST architecture

a white prescription has been acquired by the system, pre-processing steps [10] are carried out on the image in order to facilitate the subsequent OCR phase; in particular, we perform smart crop, gamma correction and image rotation.

The first operation is required since images present into our system are often simple photos provided by individuals that use their smartphone during the upload of the request of a medical service. Since the image is rarely provided by physician or other specialists, the accuracy of the prescription should be therefore improved in most cases by cropping the image to discard its negligible parts. Gamma correction [11] is usually performed in the process of digital imaging to restore as much as possible the lighting condition of the original image. Finally, image rotation is performed since it has been often generated via smartphone camera by standard users (SSN customers) so alignment could be required; to accomplish this, we exploit the barcode stored in the medical prescription, whose high contrast of black and white pixels allows a simple yet successful image alignment.

IV. TEXT EXTRACTION

After image pre-processing, text is extracted to build a string dictionary where each entry represents a field of the medical prescription. This OCR phase is carried out using EAST [12] text detector in conjunction with Tesseract [13] free OCR software. EAST is the acronym of Efficient and Accurate Scene Text Detector, based on a multi-channel fully convolutional neural network (FCN) with an efficient pipeline, whose purpose is to isolate blocks of text embedded into an image. It can be schematized as in fig. 2, where the FCN produces both information about words/phrases recognition and about the geometry of the area that contains them; both are evaluated using a threshold-based mechanism that finally provide us with text blocks to be further processed via Tesseract. Using directly Tesseract to recognize the text contained in the scanned medical prescription provided unsatisfactory results for the data set used as input, reasonably due to the low quality of canned images provided by end users. For this reason, we first used EAST, whose effectiveness in isolating text blocks was higher, then passing each block to the OCR software; this approach revealed to be slightly lower but successfully for extracting data from prescriptions. In fig. 3 is represented a scanned white medical prescription (left side), together with main text fields extracted using EAST and Tesseract (right side).

The medical prescription contains some field that are relevant for our goal, in the following briefly described:

- 'regione' stands for region (administrative area Italy is splitted into), in this case with value 'Sicily'



Fig. 3. Text extraction from a sample medical prescription

- 'assistito' is the name of end user (registered to the SSN) the medical prescription refers to
- 'indirizzo', 'cap', 'citta' and 'provincia' are different parts of the end user's address
- 'cod_fiscale' is the fiscal code (i.e. social security number) used to identify the user
- 'prescrizioni' is the list of medical services (e.g. drugs, therapies, screenings or specialist examinations as specified in section III); in the example shown, three blood tests are reported
- 'quesito' is the medical diagnosis as reported by the physician, that motivates the previous list of medical services

The last field 'quesito' is the most relevant for our purposes, since specific medical cares (field 'prescrizioni') can be allowed - and freely provided - by SSN only for specific diagnosis, therefore the field pair is used to rate the prescription as *grantable* or not, as discussed in previous sections.

V. SPELLING CORRECTION

Once text has been extracted, we proceed with a spelling correction, that is required as in most OCR softwares residual errors still occur, in particular in our scenario where the quality of scanned images is not always high (as said previously) and also the text that appears in medical prescription is usually with a reduced font size. Furthermore, the recognition of the diagnosis block ('quesito' in fig. 3) is not trivial since this field is actually a free text with variable length, hence also spelling errors due to an incorrect entry by the physician are still possible. For all these reasons, the spelling correction is applied specifically to the diagnosis block; it consists of the following tasks:

- *Non-word* error detection, that is the detection of words characterized by incorrect spelling;
- *Isolated-word* error correction, i.e. the correction of the word written incorrectly without taking into account the surrounding context;
- *Context-dependent* error correction, that is word correction characterized by spelling mistakes based on the context.

Since no specific context is provided in the medical prescription, the spelling correction algorithm we implemented focus on the first and second task listed above. To fulfill its specification, the algorithm uses a words dictionary. This

vocabulary is preliminarily obtained by extracting all the words constituting the various rules used by the system in the classification phase.

ApplySpellingCorrection method, for each word in the diagnosis block searches in the vocabulary for all words that begin with the same letter. Then the TryCorrect method is called, passing as a parameter the set of words extracted from the vocabulary. This method leverages on the Damerau-Levenshtein distance [14] to accomplish its task; such a distance is the minimum edit distance between two strings, i.e. the lowest number of character insertion, removal and/or replacement to transform the former string into the latter.

If such distance is zero the word is correct, otherwise it must be replaced with the correct word in the dictionary. In our experiments, a threshold for such distance is chosen to limit the subset of candidate words extracted from the dictionary. In the case of a null subset (for the given threshold), the word to replace is considered *unknown*, since no proper word in the dictionary has been found. The higher the threshold, the more (and possibly not suitable) words will form the subset of candidates, hence keeping as lowest as possible the value is recommended; we carried out successfully experiments using '1' as threshold.

The performance obtained from the Spelling Correction algorithms are quite satisfactory, this is due not only to the efficiency of the algorithm, but also to the use of the cache, in which all the rules are stored, the word vocabulary used for spelling correction and the different weights used to calculate the score of the different rules.

VI. INFORMATION CLASSIFICATION

The goal of information classification is to assess whether a given prescription is grantable or not, as specified in previous sections; to do this, the text extracted (and eventually corrected) is properly classified exploiting both the *Syntactic rules* and the *Rule-based tagging* NLP technique [15]. Syntactic rules are used to model all valid grammar sequences, whereas Rule-based approaches use contextual information to assign tags to unknown or ambiguous words (often called *context frame rules*). Rule-based taggers generally require supervised training, but also other approaches are available [16].

The proposed solution exhibits simplicity and good performance as it only requires the use of syntactic rules for pattern matching information extraction, and the use of rules that use data belonging to the context frame, to extract new categories of information. In this first stage of development rules and patterns have been manually built, but this time-consuming and error-prone task is going to be removed by automatized rules generation in further development of this work. The well know schema for a syntactic rule contains three tags, i.e. Source, Target and Data (see eq. 1).

$$Source \Rightarrow Target \# Data \quad (1)$$

The *Source* attribute indicates a specific pattern that the system must detect within the text string being analyzed before

it can apply the rule itself. The pattern for this attribute can be either a simple string or a syntactic expression to specify that a regular expression must be matched to detect the pattern within the text. To discriminate the type of pattern the *Source* attribute is set, rules are classified in *Regex Rule* and *String Rule*.

In a *Regex Rule* the *Source* attribute contains a Placeholder whose structure is shown in 2 and 3, where *placeholder_value_1* and *placeholder_value_n* are correct pattern matching expressions.

$$\{\{placeholder_type : *\}\} \quad (2)$$

$$\{\{placeholder_type : placeholder_value_1|placeholder_value_n\}\} \quad (3)$$

The *Target* attribute indicates the Placeholder that must be used when applying the rule to replace the pattern indicated by the *Source* attribute detected in the analyzed text. This attribute can be set in two different ways. The former requires that it simply contains the string to be used to perform the replacement, whereas the latter requires that it contains the index indicating the position of the word contained within the *Source* attribute to be used as a Placeholder when the rule is applied. In order to distinguish the two set modes, the index is always preceded by the special character £(this solution was chosen to avoid redundancy in rule coding). The replacement of pattern matching present in the rule with the value contained in the *Target* field is a text tagging operation hence the related placeholder must be characterized by a tag structure. In particular, the system provides that the placeholder of each rule is enhanced by a string having the following structure:

placeholder_type : placeholder_value

Finally, the *Data* attribute in eq. 1 indicates all the information that can be extracted from the analyzed text when the rule is applied. In general, this attribute is enhanced by a string consisting of the representation of information in JSON format. The system also allows this string to contain parameters represented in two different ways, based on their semantics. In a first case the parameter is indicated by an integer preceded by the special character &, where the integer indicates the position within the *Source* attribute of the word to be used to evaluate the parameter; another option is to use the following syntax:

index.attribute_name

where *index* indicates the position within the *Source* attribute i.e. the key that must be used to access the data structure maintained by the system containing all the information extracted through the application of the rules and *attribute_name* instead indicates the category of information of interest.

The syntax used for the coding of rules does not require that the *Data* attribute must necessarily contain a string consisting of the representation of information in JSON format. In fact, it is possible to associate the *Data* attribute with a string having the following structure: $index_0 + index_1 + \dots + index_n$

When the *Data* field receives such a value, information extracted when the rule is selected are collected from those having indexing keys equal to words from the *Source* field at position $index_0, index_1, \dots, index_n$.

The algorithm to classify information is implemented in C# and operates as follows. After rules are fetched from the configuration file, it rates each rule with a score depending on the number of words in the *Source* field, distinguishing placeholders from simple strings. In particular, the *placeholder_type* of each placeholder in the *Source* lead to a different weight for its related placeholder; such weights can be manually specified within the configuration file. Rules rating allows to establish their application order (priority).

VII. SYSTEM TESTING

In this section we describe the testing phase carried out to evaluate the performance of spelling correction and information classification algorithms, with the final purpose of establishing whether an input medical prescription can be classified as grantable or not, or eventually whether the system was not able to classify it at all.

To this purpose, a dataset with about 800.000 text rows coming from medical prescription has been used, while the rule set for the classification contains about 5000 mapping rules; note that in this test we did not consider strings extraction from medical prescription, since we focused on the assessment of performance classification. In order to efficiently perform the test, a C# script working in parallel for each row invoking an HTTP POST at the REST service was developed. During the test, several information are collected: number of traumas, number of diagnostic query, symptoms and areas present in it and, moreover, the number of spelling corrections (incorrect words and their correction and the Damerau-Levenshtein distance).

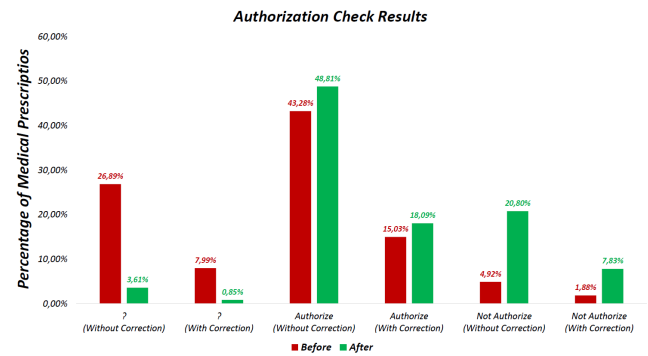


Fig. 4. Test Result

In figure 4 are shown the test results, where 'Authorize' and 'Not Authorize' indicate the grantable property, and '?' label collects those prescriptions that the system was not able to classify (named *unclassifiable* in previous sections. For each case, the two assessment (when spelling corrections are applied or not) are indicated separately.

The red line shows that approximately 30% of prescriptions are unclassifiable, a relevant (and then unacceptable) percentage in particular for the case when spelling corrections were not required. To tackle this situation, a first step was to map a lot of terms by writing a rule for each one of them, but this operation did not lead to a significant performance improvement, so we decide to include two additional information categories:

- *_si*: featuring all those words whose combination, if present within the diagnosis text block, make the prescription grantable.
- *_no*: where all those words that do not affect the classification of the medical prescription at all are stored.

The green line in figure 4 shows the result of the classification for this improved solution; in this case the classification improves significantly since only the 5% of medical prescriptions are considered unclassifiable.

VIII. CONCLUSIONS

The main goal of the proposed system is to develop a service for the analysis and authorization of medical prescriptions. Results shown that in most cases the system allows automatic classification (as grantable or not) and only 5% were not automatically classified; from tests carried out on 800,000 recipes only around 4000 therefore required manual operator intervention. The classification phase is the most relevant part of the proposed system and its quality strictly depends on the number of rules used. Their writing is a time-consuming and error-prone task, especially if manually built, therefore a planned further work is to exploit machine learning techniques to automatically manage the set of rules.

ACKNOWLEDGMENT

This work has been developed in cooperation with Previmedical s.p.a. and Previnet s.p.a

REFERENCES

- [1] V. Carchiolo, A. Longheu, M. Malgeri, and G. Mangioni, "Multisource agent-based healthcare data gathering," in *Proc. of FedCSIS*, Sep. 2015, pp. 1723–1729. [Online]. Available: <https://doi.org/10.15439/2015F302>
- [2] Y. Si and K. Roberts, "A frame-based nlp system for cancer-related information extraction." *AMIA Annu Symp Proc*, vol. 2018, pp. 1524–1533, 2018.
- [3] V. Carchiolo, A. Longheu, and M. Malgeri, "Using twitter data and sentiment analysis to study diseases dynamics," in *Proceedings of ITBAM 2015*, vol. 9267. New York, NY, USA: Springer-Verlag New York, Inc., 2015, pp. 16–24. [Online]. Available: https://dx.doi.org/10.1007/978-3-319-22741-2_2
- [4] S. Doan, E. W. Yang, S. Tilak, and M. Torii, "Using natural language processing to extract health-related causality from twitter messages," in *IEEE ICHI-W*, June 2018, pp. 84–85. [Online]. Available: <https://doi.org/10.1109/ICHI-W.2018.00031>
- [5] S. A. Parah, J. A. Sheikh, F. Ahad, N. A. Loan, and G. M. Bhat, "Information hiding in medical images: a robust medical image watermarking system for e-healthcare," *Multimedia Tools and Applications*, vol. 76, no. 8, pp. 10 599–10 633, Apr 2017. [Online]. Available: <https://doi.org/10.1007/s11042-015-3127-y>
- [6] B. Shickel, P. J. Tighe, A. Bihorac, and P. Rashidi, "Deep ehr: A survey of recent advances in deep learning techniques for electronic health record (ehr) analysis," *IEEE Journal of Biomedical and Health Informatics*, vol. 22, no. 5, pp. 1589–1604, Sep. 2018. [Online]. Available: <http://dx.doi.org/10.1109/JBHI.2017.2767063>
- [7] G. Litjens, "A survey on deep learning in medical image analysis," *Medical Image Analysis*, vol. 42, pp. 60 – 88, 2017. [Online]. Available: <https://doi.org/10.1016/j.media.2017.07.005>
- [8] S. B. Kotsiantis, "Supervised machine learning: A review of classification techniques," in *Proc. of the 2007 Conf. on EAIACE: Real World AI Systems with Applications*. Amsterdam, The Netherlands, The Netherlands: IOS Press, 2007, pp. 3–24. [Online]. Available: <https://dx.doi.org/10.1007/s10462-007-9052-3>
- [9] S. C. Wong, A. Gatt, V. Stamatescu, and M. D. McDonnell, "Understanding data augmentation for classification: When to warp?" in *2016 International Conference on Digital Image Computing: Techniques and Applications (DICTA)*, Nov 2016, pp. 1–6. [Online]. Available: <http://dx.doi.org/10.1109/DICTA.2016.7797091>
- [10] W. Bieniecki, "Image preprocessing for improving ocr accuracy," in *Intl. MEMSTECH Conf.*, 06 2007, pp. 75 – 80. [Online]. Available: <https://dx.doi.org/10.1109/MEMSTECH.2007.4283429>
- [11] X. Guan, "An image enhancement method based on gamma correction," in *2nd Intl. Symp. on CID*, vol. 1, Dec 2009, pp. 60–63. [Online]. Available: <https://dx.doi.org/10.1109/ISCID.2009.22>
- [12] X. Zhou, C. Yao, H. Wen, Y. Wang, S. Zhou, W. He, and J. Liang, "EAST: an efficient and accurate scene text detector," *CoRR*, vol. abs/1704.03155, 2017. [Online]. Available: <https://doi.org/10.1109/CVPR.2017.283>
- [13] Tesseract, "Tesseract Open Source OCR Engine," <https://github.com/tesseract-ocr/tesseract>, last accessed 08 May 2019.
- [14] F. J. Damerau, "A technique for computer detection and correction of spelling errors," *Commun. ACM*, vol. 7, no. 3, pp. 171–176, Mar. 1964. [Online]. Available: <http://doi.acm.org/10.1145/363958.363994>
- [15] E. Brill, "A simple rule-based part of speech tagger," in *Proceedings of the Third Conference on Applied Natural Language Processing*, ser. ANLC '92. Stroudsburg, PA, USA: Association for Computational Linguistics, 1992, pp. 152–155. [Online]. Available: <https://doi.org/10.3115/974499.974526>
- [16] E. Brill and M. Pop, *Unsupervised Learning of Disambiguation Rules for Part-of-Speech Tagging*. Dordrecht: Springer Netherlands, 1999, pp. 27–42. [Online]. Available: https://doi.org/10.1007/978-94-017-2390-9_3