Fuzzy Preprocessing of Medical Text Annotations of Intensive Care Units Patients

Joao Paulo Carvalho
INESC-ID
Instituto Superior Técnico –Universidade de Lisboa
R. Alves Redol 9, 1000-029 Lisboa
joao.carvalho@inesc-id.pt

Sérgio Curto
INESC-ID
R. Alves Redol 9, 1000-029 Lisboa
scurto@gmail.com

***Abstract* — Physicians' and nurses' text notes are a direct representation of experts' views on real time observable data, and contain rich information that can improve predictive medic decision making models based on physiological data. However, such notes are not usually used in such decision making models because of the dificulties in processing their content. This paper presents a new fuzzy based text preprocessing technique that allows for a more effective use of Natural Language Processing techniques when approaching medical text reports. The technique was developed to address the reports contained in the MIMIC II database.**

Keywords: *Fuzzy Text Preprocessing, Medical text reports, Natural Language Processing, MIMIC II database.*

I. INTRODUCTION

Health care decision-makers have begun to look toward engineering systems concepts for solutions to the challenging problems of improving quality and reducing costs. Several reports have highlighted the large number of preventable deaths from medical errors that occur annually, and have suggested that they result from "systems problems" [17][23]. To date, there are very few examples where engineering systems principles have been applied to the healthcare domain in order to improve quality, safety or clinical effectiveness. The exact degree to which the structure or design of a system influences outcome probably varies across clinical settings and conditions, but is likely to be significant in high-acuity, complex environments such as the intensive care units (ICU). ICUs provide a particular opportunity in the hospital to examine the benefits from implementing engineering systems approaches. Patients here are among the sickest patients in the hospital, and decisions that are made can literally mean the difference between life and death.

Patients readmitted to an ICU during the same hospitalization have an increased risk of death, length of stay, and higher costs [9][5]. Previous studies have demonstrated overall readmission rates of 4-14% [27], of which nearly a third can be attributed to premature discharge from the ICU [9]. Increasing pressures on managing care and resources in ICUs is one explanation for strategies seeking to rapidly free expensive ICU beds. Faced with this scenario, a clinician may elect to discharge a patient who has already had the benefits of stabilization and intensive monitoring, to make room for more acute patients allocated in the emergency department, exposing the outwardly transferring patients to the risk of readmission in the short term [8]. In order to allow for more objective "discharge from ICU" decisions, previous studies have presented predictive models based on physiological variables [2][3]. These models can help to base the decisions on objective criteria as well on clinical judgment.

However, none of those models makes use of the rich information contained in physicians' and nurses' text notes, which could give a better explanation of both discharges and readmissions, as they are a direct representation of the experts' views on the real time observable data. As such, they contain valuable knowledge that should/could be used to improve the results obtained using only physiological variables. This information has not been used before because of the text particularities of the documents:

- The reports are not structured as a typical written text – sentences are short, have many abbreviations, a reduced number of function words and most of the words are specific and relevant within the context;
- The reports have a large number of medical technical terms and specific technical abbreviations;
- There are many numerical values associated with physiological variables readings;
- Many different ways of expressing/representing the same information. E.g., dates (23-06-2014; 6/23/14; June, 23 2014, etc.), time (10:14PM; 22:04; 2204, etc.), etc.;
- Text is unedited, i.e., contains typographical and other word errors;
- Text contains many other artifacts, such as misplaced control characters that break sentences into paragraphs, escape sequences, etc.

These particularities prevent the effective use of common Natural Language Processing (NLP) techniques, and hinder their use as "automatic information providers". The following excerpts show examples of text extracted from ICU text reports where it is easy to understand the difficulties involved in processing and understanding the report, even for a human (note that it is impossible to show all the mentioned issues in such small excerpts):

"Pt placed on a spont breathing trial @ 13:00, pt resp one time within 10 sec -- unfortunatly his SBP droppd from 100 to 70 rapidly and therefore the trail was d/c'ed."

"Cardiac: BP stable 120-130/60. Pt is on Amiodarone via NGT TID. Tolerating this well. HR 80-95 most of the shift. Has rare to occ. PVC/APC. Swan numbers done Q6hrs as ordered and probably swan will come out today. CVP 7-9, PCW 16-20, CO [**6-2**] and SVR 800-900. He remains on heparin drip which needed to be decreased to 750u/hr at 11PM for PTT 110. Repeat PTT will be sent at 5AM."

In this paper we propose a new fuzzy based text preprocessing technique that allows for a more effective use of NLP techniques when approaching medical text reports, and show examples of how the resulting preprocessed texts are ready for the application of bag of word text processing related techniques such as, for example, the creation of fuzzy fingerprints of likely to be readmitted patients [10][11]. Bag-of-words is a simplified model used in natural language processing and information retrieval. In this model, a text is represented as an unordered collection of words, disregarding grammar and even word order. The simplifying assumption that those variables are independent is considered. Since these techniques rely essentially on word and/or feature counts, the indicated medical annotations text issues become extremely relevant due to the fact that different representations of a feature should all be accounted as the same feature.

The paper is organized as follows: Section II, introduces the database used in this work. Section III briefly mentions some related work and the specifics of the work here proposed. Section IV describes the proposed Fuzzy Text Preprocessing. It is divided into several subsections described the several phases of the process. Each subsection includes examples, some results and future work. Finally section V presents some more results and the main conclusions regarding the developed work.

II. THE MIMIC II DATABASE

The developed work uses data from the Multi-parameter Intelligent Monitoring for Intensive Care (MIMIC II) database [21]. This is a large database of ICU patients admitted to the Beth Israel Deaconess Medical Center, collected from 2001 to 2006, and that has been de-identified by removal of all Protected Health Information. The MIMIC II database is currently formed by 26,655 patients, of which 19,075 are adults (>15 years old at time admission). It includes high frequency sampled data of bedside monitors, clinical data (laboratory tests, physicians' and nurses' notes, imaging reports, medications and other input/output events related to the patient) and demographic data. From the available data, we focused mainly on the physicians' and nurses' notes, but we also used patient relevant data such as, for example, age.

III. RELATED WORK

It is possible to find in the literature several works that address the use and the processing of medical texts to extract information. However most works are related with standard medical texts (such as textbooks or essays), and therefore are of very limited application to the problem we address here [19][6]. Other works, such as [28][29] could simply not be applied to the current problem without the prior preprocessing here proposed. To our knowledge no work exists to fully address medical text databases as extensive, complex and problematic as MIMIC II.

The current work involves techniques from areas such as Natural Language Processing (NLP), Machine Learning, and Computational Intelligence. References to the used techniques and technical terms are given throughout the text wherever they are needed.

IV. FUZZY TEXT PREPROCESSING

Text preprocessing is usually known as the task of converting a raw text file, essentially a sequence of digital bits, into a well-defined sequence of linguistically meaningful units [24]. Text pre-processing is an essential part of any NLP system, since the characters, words, and sentences identified at this stage are the fundamental units passed to all further processing stages. The proposed text preprocessing differs in some aspects to standard text preprocessing due to the main intended application: prepare the text to be used in bag of word techniques. The proposed fuzzy text preprocessing is roughly divided into the following phases: Text cleaning; Word error correction; Abbreviations' processing; Normalization of date and time formats; Normalization of units; Text fuzzification; and finally, preparation for text extraction.

A. Text cleaning

Text reports present in the MIMIC II database were obtained and created using several workflow systems: some originate from Windows based computers, others LINUX, others MacOS; some were digitized, others typed; the original file format is widely variable (.txt, .doc, etc.); and so on. This originates several problems in the raw text that must be addressed by what we refer to as "text cleaning". This preprocessing phase roughly consists in a set of character manipulation operations (insertion / removal / replacement) necessary to correct formatting, encoding and minor typing errors, in order prepare the text for further operations. The semantic information contained in the original text should not suffer changes. The database text was sampled, and the samples analyzed in order to find what operations were deemed necessary. This phase was divided into several stages, where the order of stage execution is relevant since it influences the outcome. The following stages were defined:

- New lines removal: The text in the MIMIC II reports contains lots of padding new lines that break paragraphs. In order to recover the sentences, such new line characters have to be ignored.
- Whitespaces normalization: Replacement of all occurrences by a single space. Reports often contain different types of whitespaces (tabulations, non-breaking spaces, etc.) that do not add any information to the text.
- Backslash characters '\' corrections: Replacement of multiple backslash characters with a single one. The repetition of backslash characters, probably used to escape the text, does not add information and in some cases increases the difficulty of extracting fractions and abbreviations.

- Punctuation normalization: Replacement of four or more occurrences of the same punctuation character with only three instances.
- Repeated letters normalization: Replacement of four or more occurrences of the letter with only three instances. Attempts to work around typos or emphasis on a specific letter (may break acronyms).
- Insertion of space characters in the following cases:
    - Between MIMIC II special markings "[**<text>**]" and the text contained in them. Tokenization (section IV.G) cannot work properly when there is no separation between the '*' and the <text> because this is not a typical usage for this characters on the used parsers' training set.
    - Between '#' and the text that follows it. In most modern texts, the meaning of a '#' immediately followed by text is a "hashtag". However, in the MIMIC II database this is the result of a typing error: it usually represents a bullet point where the spacing between the bullet and the text was not typed.
    - Between slash characters and the rest of the text (except with numbers): These characters are often used to enumerate combinations or alternatives (medicines, treatments, issues, etc.). This operation allows us to count and process each word by itself instead of working with huge tokens that cannot be compared with the rest of the reports
    - Between numbers and alphabetical characters (except on tokens composed by numbers followed by st, nd, rd and th). This step is an attempt to normalize numeric quantities associated with a unit, since in some instances there are spaces separating the values and the corresponding units. Special care was taken to keep numerals as is (1^{st}, 2^{nd}, …). This step obviously has the harmful effect of breaking codes into multiple tokens.

The text clean up is performed using regular expressions that were specifically developed to address the text in the MIMIC II database. A regular expression [4] is a sequence of characters that forms a search pattern, mainly for use in pattern matching with strings (string matching), i.e. "find and replace"-like operations.

Next we present some examples of the simpler regular expressions used in the clean up step:

- newline: "(?m)[\r\n\u0085\u2028\u2029]+". Captures all characters that might be considered a "new line", such as: \r - carriage return; \n – newline; \u0085 - next line; \u2028 - line separator; \u2029 - paragraph separator
- spacing characters: "(\s|\u00A0)+". Normalizes characters representing spaces used to remove multiple occurrences, such as: \s - a Java regular expression addressing whitespaces [\t\n\x0B\f\r]; \u00A0 – a non-breaking space character not considered in Java as a white space (in html:)
- Processing of MIMIC2 anonymization markings in order to facilitate extraction of dates, physicians ID's, x-ray reports ID's, etc.: "(\[**)([^*]+)(**\])". Example: change "[**3399-4-29**] 11:31 PM" into "[** 3399-4-29 **] 11:31 PM"

The previously described operations allowed for a huge advance in what concerns the usability of the MIMIC II text database. However it is still possible to detect instances where the text-cleaning phase is not yet 100% satisfactory, so the regular expressions are still being updated and improved.

B. Automatic Fuzzy Word Error Detection and Correction

The second preprocessing step consists in automatic word error detection and correction, using a novel procedure and an efficient fuzzy word similarity function previously presented [7].

Word error detection and correction in unedited technical text is a complex and expensive task that must usually be done manually, or, even if done automatically, demands a significant human intervention. In our case, this task must be somehow automated since the size of the MIMIC II database would make the cost of manual offline text editing unbearably expensive. One might ask why can't a simple dictionary-based spelling correction tool be used? There are several motives that prevent such use: most automatic spelling tools often need a rather intensive human input; automatic spelling correction is strongly limited by the number of word errors (one or at most two), even if many words contain more than a single typographical error; many words present in MIMIC II database, such as technical terms or abbreviations, are not present in dictionaries and as such, not recognized; etc.

The MIMIC II database contains a total of 156 million words with 3 or more characters, and 260180 distinct words (a rather huge number). Of these 260180 distinct words, only 31527 (12%) appear in known word lists: 30828 appear on the SIL list of English known words (which contains 109582 distinct words) [16], and 429 appear on an additional list containing medical terms not common in English. The remaining 228923 words are simply unknown to dictionaries, and most are the result of typing errors!

As an example of the extent of such typing errors, here is a non-extensive list of the different misspelled variants of the word "abdomen" found in the MIMIC II database: abadomen, abdaomen, abndomen, badomen, abdaomen, abdeomen, abdcomen, abdemon, abdeom, abdoem, abdmoen, abdemon, abdiomen, abdman, abdmen, abdme, abddmen, abbomen, abdmn, abdme, abdmonen, abdonem, abdoben, abdodmen, abdoemen, abdoem, abdoem, abdomin. Just out of curiosity, the incorrect form "abdomin" appears 1968 times in the database.

Given the above considerations, and the impact of such errors when considering any kind of text analysis, we developed an automatic procedure to detect and correct typographical and other word errors in the MIMIC II text corpus. The automatic corrected-word list creation proceeds as follows:

1. Count and create a list with all words in the corpus data.

2. Order the list based on most frequent words.
3. Create a list of unknown words by comparing all words with an extensive known word list composed by the SIL English word list and the medical terms list.
4. Create two new word lists, top-*k* and bottom-*l*, containing respectively the *k* most frequent unknown words and *l* least frequent unknown words. We use the assumption that the top-*k* list should not contain words with typographical errors – the same word would have to be almost constantly miswritten (and the error would have to be exactly the same) for it to appear in top-*k*. Around 80% of the unknown words occur 5 times or less in the MIMIC II database. So we are empirically using this value to define the bottom-*l* size.
5. Use a word similarity function to compare each word in the bottom-*l* list of unknown words with each word in the top-*k* list (in order to find misspelled technical terms that are not known), and then with each word in the extensive known word list. Unknown words are corrected to the most similar word as long as the similarity is above a given threshold.
6. Words that are not paired can be left as is, since bottom-*l* words that are not related to the top-*k* words should not be relevant to our problem.

In order for the above procedure to give good results, a good word similarity function that is especially adapted to deal with common typographical errors is needed. Current research on string similarity offers a panoply of measures that can be used in this context, such as the ones based on edit distances or on the length of the longest common subsequence of the strings. However, most of the existing measures have their own drawbacks. For instance, some do not take into consideration linguistically driven misspellings, others the phonetics of the string or the mistakes resulting from the input device. Moreover, the majority of the existing measures do not have a strong discriminative power, i.e., both a good precision and recall, and, therefore, it is difficult to evaluate if the proposed suggestion is reasonable or not, which is a core issue in unsupervised spelling.

Therefore we developed a novel word similarity function, the Uke Word Similarity (UWS), and its fuzzy variant, FUWS [7]. This word similarity function combines the most interesting characteristics of the two main philosophies in word and string matching, and by integrating specific fuzzy based expert knowledge concerning typographical errors, can achieve a good discrimination. The similarity threshold value used to process the MIMIC II database was 0.67, as suggested in [7].

It is obviously impossible to check the validity of all 228923 potential corrections, but a small result sample shows that around 90% of the corrections are appropriate. If these results are generalizable, the number of correct distinct words increases from 12% to more than 90%.

Some of the words are not corrected using the proposed method because they are so frequent that they are not included in the bottom-*l* list. A good example is the word "tolerated", which is wrongly written "toelrated" 306 times. In order to automatically detect such cases, we plan to make a second pass checking all unknown words against all known words, and correcting those whose FUWS similarity is above 0.90. This should still capture these common mistakes while not correcting medical technical terms such as "venographic" which also occurs 306 times.

C. Dealing with Abbreviations

The third preprocessing step consists in dealing with the large amount of abbreviations one can find in MIMIC II text database. The use of medical abbreviations is not a problem *per se* if one assumes that the reports are to be read and interpreted by physicians, and they should be familiar with them. The problem lies in the fact that the use of the abbreviations and/or the name is not consistent – different abbreviations for the same term or a mix of abbreviations and terms are common. Since both refer to the same information, the text should be made consistent. In order to solve this issue a list/dictionary of medical abbreviations is being compiled based on information extracted from [14][18]. This list is used to replace abbreviations found in the database by the corresponding word / medical term. Alternatively it is also possible to normalize all representations into commonly accepted abbreviations. For example, the text:

"No ECG changes, but after 46 seconds his BP dropped",

can be normalized to either:

"No electrocardiogram changes, but after 46 seconds his blood pressure dropped",

or to:

"No EKG changes, but after 46 seconds his BP dropped".

On top of the use of regular medical abbreviations, the text reports in the MIMIC II database often contain non-technical abbreviations such as, for example "pt" instead of patient, or "d/c'ed" instead of "disconnected". A list/dictionary of commonly found abbreviations is also being created from sampled reports.

D. Normalization of Units

Since the main goal of the MIMIC II data base text processing is to gather relevant information that can be used to give a better explanation of the evolution of the patients medical condition while their stay in the ICU, and find the motives that lead to a readmission shortly after discharge from ICU, it is also important to normalize the medical information contained in the reports. The first step in this process consists in guaranteeing that the units used when describing patient physiological measures, or data such as medical prescriptions, is consistent. We found out that the International System of Units (SI) standard is not consistently followed in the reports. This means that it is possible to find the same information expressed in different forms. For example, in a given ward a patient's weight can be given in pounds, and when moved to a new ward it might be given in kilograms. Additionally it is possible to find cases where a patient is measured in pounds, and then medication is given based on his weight in kilograms. For example (extracts):

Report a: "…Weight preoperatively was 175 pounds…";

Report b (referring to the same patient): "…total fluids of 150cc per kg per day…";

Report c (still the same patient): "…The patient should be weighted daily and an increase of more than 3 pounds per day should be reason for reassessment…".

The procedure to normalize units is based on a "medical terms/units" dictionary that is currently under construction, and proceeds is as follows: 1) Search for a token containing units (e.g. mmHg); 2) Find the "feature" (physiological measure, medication, etc.) associated with those units; 3) Look in the dictionary for the corresponding SI units; 4) Convert the value associated with the units to the corresponding SI units (if necessary).

Following this procedure, the previous example is processed to:

Report a: "…Weight preoperatively was 79.4 kg…";

Report b: "…total fluids of 150 cc/kg/day…";

Report c: "…The patient should be weighted daily and an increase of more than 1.4 kg/day should be reason for reassessment…".

One related issue to be solved lies in the fact that sometimes units are omitted. In such cases it is not trivial to execute the algorithm since steps 1 and 2 must be replaced by a more complex procedure that is still in early development. The procedure is based in the observation that units are usually referred to at least once in each report.

E. Normalization of Date and Time Formats

Another important preprocessing step is the normalization of Date and Time formats, since in a sequence of reports, the same physiologic data may have different values measured at different occasions as the patient progresses through ICU wards. Since this timing information might come from different (and unstructured) reports, it may be represented in a number of different formats. For example, to represent an event that started at 10am in different reports we might have: 1000; 10am; 10:00; 1000-1400; 10hrs.

To ease the text extraction phase, all the dates and time references should use a consistent representation. To accomplish this, the references captured by our regular expressions are converted to the following format:

"4 digits year" "-" "2 digits month" "-" "2 digits day" "SPACE" "2 digits 24-hours format" ":" "2 digits minutes" ":" "2 digits seconds"

When one of the values is missing the character '_' replaces it. Examples:

- "January 27th, 2013 at 10pm" is converted to "2013-01-27 22:00:00";
- "At 10pm" is converted to "____-__-__ 22:__:__".

Regular expressions were developed to perform the date and time normalization. These expressions are rather long and complex. Following are examples of partial extracts of the regular expressions used to detect:

a) Months represented in textual form:

"(?:January|Jan\\.?|February|Feb\\.?|March|Mar\\.?|April|Apr\\.?|May|June|Jun\\.?|July|Jul\\.?|August|Aug\\.?|September|Sep\\.?|October|Oct\\.?|November|Nov\\.?|December|Dec\\.?)";

b) Several forms for representing days:

"(?i)(?:(?:[1-3]?(?:0th|1st|2nd|3rd|[49]th)|11th|12th|13th)|(?:3[01]|[0-2]{0,1}\\d)|(?:(?:(?:twenty\\s|thirty\\s)?(?:first|second|third|fourth|fifth|sixth|seventh|eighth|ninth))|tenth|eleventh|twelfth|(?:thir|four|fif|six|seven|eigh|nine)teenth)|twentieth|thirtieth)".

F. Text fuzzification

The proper application of the previous pre-processing steps allows us to obtain a cleaner and consistent medical text database that is essentially ready for NLP and for the application of classification techniques. However, if the techniques to be applied rely heavily on word or feature counts, as it is common on, for example, bag of words based techniques, then not much could be learned from it. This is due to the fact that numerical information can be too sparse to perform proper classification given the number of relevant cases and the large number of possible features. This can be explained using a naive example. Let us assume 5 patients, with systolic blood pressure (SBP) values of, respectively, 183mmHg, 189mmHg, 181mmHg, 190mmHg and 187mmHg. Now let us assume that all 5 patients were discharged the day after those values were measured, and that all were readmitted to the ICU after less than 24 hours. Let us also assume that no other patients with SBP values above 180 were discharged the day after the measurement was taken. The obvious conclusion would be that patients with SBP above 180mmHg should not be discharged the following day. However, the direct application of bag of words' methods would not be sufficient by itself to extract this information, since there would be only one count for SBP=181, one for SBP=183, one for SBP=187 and one for SBP=190. These results would have to be somehow categorized *a posteriori* to extract such conclusion. Here we propose a preprocessing method that allows for this kind of numerical information to be directly taken into account by bag of words' methods. Note that this would not be so relevant when applying classification techniques such as Support Vector Machines (SVM), since such techniques can cope with this problem quite well. However, bag of word techniques can perform better than such techniques for text analysis [10][26], especially in what concerns computational performance, which is a crucial factor in future applications.

The method consists in fuzzifying ngrams, i.e., sequences of *n* relevant tokens (words). Continuing the previous example, the text "patient SBP was 183 mmHg", containing the 4gram "SBP was 183 mmHg", would be fuzzified according to appropriate fuzzy linguistic membership functions, into "SBP was Hypertensive_Emergency". The same would happen to the remaining 4 cases, and as such, there would be a count of 5 Hypertensive_Emergency cases, and all were readmitted to the ICU.

TABLE I shows a classification for blood pressure in adults [1], and Figure 1 shows the developed fuzzy linguistic terms to represent Systolic Blood Pressure.

TABLE I: Blood pressure classification in adults

Category	systolic mmHg	diastolic mmHg
Hypotension	< 90	< 60
Desired	90–119	60–79
Prehypertension	120–139	80–89
Stage 1 Hypertension	140–159	90–99
Stage 2 Hypertension	160–179	100–109
Hypertensive Emergency	≥ 180	≥ 110

When the numerical values have a non-zero membership degree in two linguistic terms, both are indicated. E.g., "SBP was 178mmHg", is represented as "SBP was S2Hypertension / Hypertensive_Emergency" (see Figure 1). Note that the original values, as well as the calculated membership degrees, are kept in the database in case they are necessary for more detailed analysis.

Since some membership functions are dependent on the patient age, as is the case, for example, of heart rate, the fuzzification of such physiological variables takes into account the patient age.

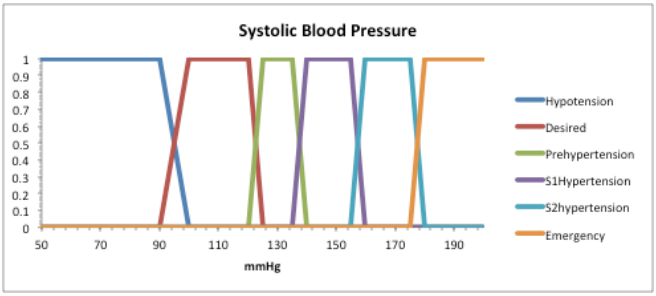

Figure 1 – Systolic Blood Pressure Fuzzification.

The process to capture the relevant ngrams uses regular expressions and is highly simplified by the previous normalization and abbreviations' preprocessing (sections IV.C, IV.D and IV.E). So, the main work consists in the definition of the linguistic membership functions. This is an ongoing work due to the high number of possible physiological variables and the involved expert knowledge. Up to now, we have defined and implemented the procedure for the following variables: Age; Blood Pressure (systolic and diastolic); Creatinine; Heart Rate; Oxygen Saturation; Respiratory Rate; Temperature; Weight; Blood Cell count.

G. *Text Extraction*

In order to do eventual semantic analysis, it is necessary to prepare the data base for text extraction, so we also performed tokenization, segmentation, tree parsing, and dependency parsing on the MIMIC II text database. Tokenization is the process of breaking a stream of text up into words, phrases, symbols, or other meaningful elements called tokens. The list of tokens becomes input for further processing such as parsing or text mining. Text segmentation is the process of dividing written text into meaningful units, such as words, sentences, or topics. A parsing tree is an ordered, rooted tree that represents the syntactic structure of a string according to some context-free grammar. Dependency parsing is a procedure that finds a one-to-one correspondence between every element in a sentence and one node in the structure of that sentence that corresponds to that element.

Several tools and techniques were combined and tested to find the most adequate procedures considering the particular characteristics of the MIMIC II texts. The following tools were used:

- Tokenizers: L2F PTBTokenizer, Lingpipe [12] and Stanford [15][20];
- Segmenters: Lingpipe [12] and Stanford [15][20];
- Tree parser: Stanford [15][20];
- Dependency parser: Stanford [22].

The L2F PTBTokenizer is a modified version of the Berkley Parser PTBLexer [13][25].

The L2F PTBTokenizer and Lingpipe combinations are faster, but the speed difference is irrelevant in the cases where the information extracted through parsing and/or dependencies is important for final prediction models, since the Stanford CoreNLP chain can do all the tasks in a single annotation task (it does all the tasks up to Dependency parsing, unless they are explicitly disabled, since each requires the previous tokenization > segmenting > parsing).

The best empirical results so far use the full Stanford options. The alternative is the L2F PTBTokenizer / LingpipeSegmenter / StandfordParsers combo.

V. Results and Conclusions

This paper presents a new fuzzy based text preprocessing technique that allows for a more effective use of NLP techniques when approaching the MIMIC II text database. As indicated throughout the paper, this is still an ongoing work, and despite some very interesting and promising results, there is still plenty to do. However, the pre-processing of the MIMIC II text database is an incremental work. As such, the existing ongoing preprocessed text, which results from the already implemented operations, can advantageously be used for information extraction instead of the original. The results are obviously not perfect since none of preprocessing steps is error free: some regular expressions still fail to capture all the instances; not all abbreviations are normalized; the accuracy in word error correction is not 100%; etc. One cannot either ignore the fact that the operations are sequential, and that errors in the early phases are propagated throughout the whole process. However, since (and as long as) no major technical errors are committed in each of the steps, the ongoing results represent a large improvement over the original unprocessed database. Following is a small example of the current preprocessing results for the texts presented in section I:

"Patient placed on a spontaneous breathing trial @ ____-__-__ 13:00:__, patient responds one time within ____-__-__ __:__:10 – unfortunately his systolic blood pressure dropped

from Desired to Hypotension rapidly and therefore the trail was disconnected."

"Cardiac: blood pressure stable Desired / Desired/Hypotention. Patient is on Amiodarone via nasogastric tube three times a day. Tolerating this well. Heart rate Slight Acceleration most of the shift. Has rare to occ. Premature ventricular contractions / Premature atrial contractions. Swan-Ganz catheter numbers done every ____-__-__ 06:00:00 as ordered and probably Swan-Ganz catheter will come out today. Central venous pressure 7-9 mmHg, Pulmonary capillary wedge 16-20 mmHg, Cardiac output [** 6-2 **] and Systemic vascular resistance 80-90 MPa.s/m3. He remains on heparin drip which needed to be decreased to 750 units/hr at 2004-02-21 23:00:00 for Partial Thromboplastin Time 110 s. Repeat Partial Thromboplastin Time will be sent at 2004-02-21 05:00:00.

The first example was preprocessed without issues: some typing errors were corrected; all abbreviations were recognized; time and dates were normalized properly; the measured values were all fuzzified. The second example preprocessing is not as complete since most of the identified physiological variables are yet not prepared for fuzzification (only heart rate was fuzzified). It is also possible to detect an issue with the date/time normalization, since the preprocessing fails to detect the change of day in the final line (however this is harmless in this example) This could be eventually improved in the future by enhancing the normalization process. All the remaining steps performed quite well, especially in what concerns abbreviations, and normalization and conversion of units. Only "occ" failed to be processed.

Overall, it is possible to say that the results are very interesting and devoid of errors that hinder further processing.

Finally, as a result of the current fuzzy preprocessing, we can also obtain separated text reports dedicated to each of the variables that have already been fuzzified. These reports can be indexed by (anonymized) patient id, age, date and report id. With such reports it is already possible to perform several classification tasks using techniques such as Fuzzy Fingerprints [10][26] that can be used to improve results of the predictive models based on physiological variables readings [2][3].

ACKNOWLEDGMENT

This work was supported by national funds through FCT – Fundação para a Ciência e a Tecnologia, under project PTDC/EMS-SIS/3220/2012 and project PEst- OE/EEI/LA0021/2013.

REFERENCES

[1] "Understanding blood pressure readings", American Heart Association. 11 January 2011, last accessed January 2014.

[2] A. S. Fialho, F. Cismondi, S. M. Vieira, S. R. Reti, J. M. C. Sousa, and S. N. Finkelstein, "Data mining using clinical physiology at discharge to predict icu readmissions," *Expert Systems with Applications*, vol. 39, no. 18, pp. 13 158–13 165, December 2012.

[3] A. S. Fialho, U. Kaymak, F. Cismondi, S. M. Vieira, S. R. Reti, J. M. C. Sousa, and S. N. Finkelstein, "Predicting intensive care unit readmissions using probabilistic fuzzy systems," Proc. of FUZZ-IEEE 2013, Hyderabad, India, 2013.

[4] Aho, A.V., "Algorithms for finding patterns in strings". In van Leeuwen, Jan, Handbook of Theoretical Computer Science, volume A: Algorithms and Complexity, pp. 255–300, 1990 The MIT Press.

[5] Boudesteijn E, Arbous S, van den Berg P. Predictors of intensive care unit readmission within 48 hours after discharge. Critical Care 2007, 11(Suppl 2):P475

[6] Cao, C., "Extracting and sharing knowledge from medical texts", Journal of Computer Science and Technology, Volume 17, Issue 3, pp 295-303, 2002, Springer.

[7] Carvalho, J.P., Coheur, L., "Introducing UWS – A Fuzzy Based Word Similarity Function with Good Discrimination Capability: Preliminary results", Proc. of the FUZZ-IEEE 2013, Hyderabad, India, 2013.

[8] Chalfin DB, Trzeciak S, Likourezos A, et al. Impact of delayed transfer of critically ill patients from the emergency department to the intensive care unit. Crit Care Med. 2007; 35(6):1477-83.

[9] Durbin CG Jr, Kopel RF. A case-control study of patients readmitted to the intensive care unit. Crit Care Med. 1993 Oct;21(10):1547-53

[10] Homem, N. Carvalho, J.P., "Authorship Identification and Author Fuzzy Fingerprints", Proc. of the NAFIPS2011 - 30th Annual Conference of the North American Fuzzy Information Processing Society, 2011, IEEE Xplorer

[11] Homem, N. Carvalho, J.P., "Finding top-k elements in data streams", Information Sciences, 180(24), pp. 4958-4974, Dec. 2010, Elsevier.

[12] http://alias-i.com/lingpipe/

[13] http://code.google.com/p/berkeleyparser/

[14] http://en.wikipedia.org/wiki/List_of_medical_abbreviations

[15] http://nlp.stanford.edu/software/lex-parser.shtml

[16] http://www-01.sil.org/linguistics/wordlists/english/, last accessed on February 2014.

[17] Institute of Medicine, "Crossing the Quality Chasm: A New Health System for the 21st Century". National Academy Press, pp. 1- 22, Executive Summary, (2001).

[18] JD.MD, Inc. online Medical & Dental Abbreviations Glossary, http://www.jdmd.com/abbreviations-glossary.asp, last accessed on 7/3/2014.

[19] Khoo, C.S., Chan, S., Niu, Y., "Extracting Causal Knowledge from a Medical Database Using Graphical Patterns", proc. of the 38th Annual Meeting of the ACL, pp.336-343,Hong Kong, 2000, ACL

[20] Klein, D., Manning, C.D., "Fast Exact Inference with a Factored Model for Natural Language Parsing." In Advances in Neural Information Processing Systems 15 (NIPS 2002), pp. 3-10, 2003, Cambridge, MA: MIT Press

[21] M. Saeed, C. Lieu, and R. Mark, "Mimic ii: A massive temporal icu database to support research in intelligence patient monitoring," *Computers in Cardiology*, vol. 29, pp. 641–644, 2002.

[22] Marneffe, M.C:, MacCartney, B., and Manning, C.D., "Generating Typed Dependency Parses from Phrase Structure Parses". In proc. of the LREC 2006.

[23] National Academy of Engineering and Institute of Medicine, "Building a Better Delivery Sytem: A New Engineering/Health Care Partnership". National Academies Press, pp. 1-26, (2005).

[24] Palmer, D., "Text Pre-processing", Handbook of Natural Language Processing, Second Edition, CRC Press, Taylor and Francis, 2010.

[25] Petrov, S., Klein, D., "Improved Inference for Unlexicalized Parsing", Proceedings of HLT-NAACL 2007, pp. 404-411, Rochester, NY, 2007, ACL.

[26] Rosa, H., Batista F., Carvalho, J.P., "Twitter Topic Fuzzy Fingerprints", Proc. of the 2014 IEEE World Congress on Computational Intelligence, WCCI2014, Beijing, China, 2014.

[27] Rosenberg AL, Watts CM: Patients readmitted to intensive care units: A systematic review of risk factors and outcomes. Chest 2000; 118:492-502

[28] Weeber, M., Vos, R., "Extracting Expert Knowledge from Medical Texts", available online http://citeseerx.ist.psu.edu/viewdoc/download?rep=rep1&type=pdf&doi=10.1.1.45.7749, last accessed January 2014.

[29] Zhou, L. et al, "Using Medical Text Extraction, Reasoning and Mapping System (MTERMS) to Process Medication Information in Outpatient Clinical Notes", AMIA Annu Symp Proc. 2011; 2011: 1639–1648.