# ECNU at 2018 eHealth Task1 Multilingual Information Extraction

Mengting Li[1], Cong Xu[1], Tingyu Wei[1], Dongyang Bao[1], Ningjie Lu[1], and Jing Yang[1,2]

[1] East China Normal University, Shanghai 200062, China
{51174506015,51174506089,51174506021,51174507001,51174506094}@stu.ecnu.edu.cn
[2] Shanghai Key Laboratory of Multidimensional Information Processing
jyang@cs.ecnu.edu.cn

**Abstract.** The CLEF eHealth 2018 Task 1 is aimed to automatically assign ICD10 codes to the text content of death certificates. The challenges of this task is that participants have to extract information from written text in unexplored French language corpora, which means that all these ICD10 codes have little data used to train. In this paper, our team proposes some methods to solve the Task 1. We utilize two machine learning method, Xgboost and RandomForest, meanwhile, we also take advantage of some association rules and similarity computation to boost the performance of our method. We evaluate our results using the evaluating code provided by organizer.

**Keywords:** Xgboost · RandomForest · Regular Match Expressions · Similarity Computation · Information Extraction · Text Classification.

## 1 Introduction

Based on the pre-work of the 2016[1] and 2017[2] tasks which already addressed the analysis of French biomedical text with the extraction of causes of death from a corpus of death reports in French[3], the goal of CLEF eHealth 2018 Task 1 is to automatically assign ICD10 codes to the text content of death certificates [4, 5] . The ICD10 codes are divided into 26 alphabetic classes, such as A, B and Z, besides, there are also some digital coding behind the alphabetic coding. Therefore, we can regard this task as a multi-classification.

The data set is called the CepiDC Causes of Death Corpus, it comprises free-text descriptions of causes of death which are reported by physicians in the standardized causes of death forms. The training data have two types: raw data and aligned data. Raw data contains 65,843 death certificates, and different files have different information, such as DocID, RawText, Gender, Age and so on. Different from raw data, aligned data combines causes information, identification and cause labels together. Therefore, every case in aligned data has complete information, and the last three fields are CauseRank, StandardText and ICD10, which will not exist in test data.

| Index | Text |
|---|---|
| A | septicémie streptocoque B hémolytique septicémie streptocoque B septicémie streptocoque alpha-hémolytique ... |
| B | septicémie streptocoque septicémie staphylocoque cathéter dialyse ... |
| J | sepsis ORL sepsis origine pulmonaire sepsis médiastino-pleural sepsis médiastinal ... |
| I | sénilité vasculaire sénilité cardio-vasculaire sénilité cardiaque ... |
| Z | sédation antiépileptique sédation antidouleur sédation antalgique ... |
| Total | 26 |

Table 1: Dictionaries

In this task, the organizers have provided vague tag and the causes of death are all medical vocabulary. The ICD10 codes, such as S299, V892 and I259, are in the same category. In this way, it is less useful to classify those data depending on semantic information. In order to obtain accurate results, we propose two methods: Regular Match Expressions and hybrid approach based on machine learning. We should assign one or more ICD10 codes to each cause of death (one case may contain several ICD10 codes). In one file, there are more than 3,000 ICD10 codes. The dictionaries are summarized in table 1(6 versions of a manually curated ICD10 dictionary developed at CépiDC).

Our system architecture is showed in Figure 1. We mainly utilize the Regular Match Expressions to obtain the results. In order to handle those data which are unable to find out the mapping expressions, we extract some features from training data and utilize machine learning method to classify. To improve the accuracy rate, we design a strategy to pre-classify these aligned data into 26 tags whose range is from A to Z and those tags represent the first code of ICD10 codes. Furthermore, we apply similarity computation and regular match expressions to obtain the digital code behind the alphabet code. At last, we combine the results of machine learning classification and regular classification as the final results.

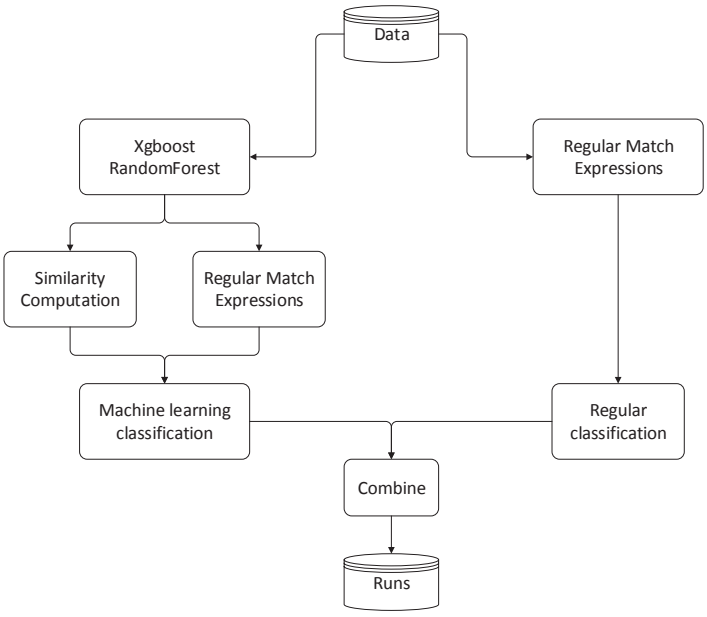


Fig. 1: Framework Structure of the System

## 2 Challenge

The language of data is French, therefore, it is so difficult for us to understand those causes of death perfectly. According to the requirement of this task, we are prohibited from using translation tools, thus, what we do is try our best to extract the features and inner connection between data. And the majors of the members of our team are all computer science, so we know little about medical knowledge. What's more, in the case of data itself, the difference between different categories is not obvious.

## 3 Methodology

We extract raw text, standard text and dictionaries provided by the organizers in training data as gold data. We divide gold data into 26 sets {set_A, set_B, ..., set_Z} according to the algebraic code, for example, the raw text set is {S1, S2, S3, S4, S5} and the corresponding ICD10 code set is {A123, A234, D135, X246, A145}, therefore, {S1, S2, S5} belongs to set_A and {S3} belongs to set_D, and {S4} belongs to set_X. Besides, we use Google's word2vec model to train the whole gold data to get the vec.bin file in order that we have an access to compute similarity between training data and test data. We extracted the ICD10 corresponding code from the training data provided by the organizer and extracted all the raw text and aligned texts as corpus, and then directly used

google's word2vec model to train our own corpus. At the same time, because of the grammatical and formal differences between French and English, we need to do some preprocessing, such as extracting stems and removing stop words. In addition, we did not consider the difference between English and French too much, because word2vec does not need any known semantic knowledge during training, and it completely depends on the corpus.

### 3.1 Regular match expressions

We regard gold data as regular expressions to match, so if the text $T$ in test data directly appears in gold data, we predict that the ICD10 code of $T$ is the corresponding code. However, sometimes $T$ may contain two or more kinds of causes of death, in this case, we will split $T$ according to expressions. For instance, $T$ is "Tableau de mort subite au cours d'un effort sportif" and regular expressions $R$ contains $R_1$"mort subite" and $R_2$"effort sportif". When we use $T$ to match $R$, we will find that $R_1$ is included in $T$, so we will predict that the ICD10 code of $T$ is the same with $R_1$. This method ignores $R_2$ and lead to an incomplete prediction, so we will split $T$ into two parts, $T_1$" mort subite" and $T_2$" Tableau de au cours d'un effort sportif", and use $T_2$ to match $R$ and split $T_2$ until there is no more match between $T_n$ and $R$. In the end, the ICD10 codes of $T$ is a collection of all $T_i$'s mapping codes.

Because there are two or more causes of death in a raw text, we adopted an iterative strategy to improve the accuracy of the regular match: For $S$ belonging to the raw text set, we set it with the regular expression set $R = \{r_1, r_2, ...r_n\}$ to match, when a certain expression $r_i$ is matched in $S$, $r_i$ is extracted from $S$ as $T_1$, and the remaining element in $S$ is taken as $S$; judging whether $S$ is empty, if it is not empty, it continues to match $R$ and split it and record $r_i$ as $T_i$,until the final phrase or word(entities) $S$ can't find matching elements in $R$. At this time we consider the match is over.

### 3.2 RandomForest and Xgboost method

We utilize two kinds of machine learning methods: bagging and boosting. RandomForest[6] [7] belongs to bagging and Xgboost[8] [9] belongs to boosting. In order to train models, we extract some features from training data, which are listed in table 2. All these features are provided by the organizers in training files. What's more, we use some natural language processing tools to extract semantic features, such as stop words and stems. Our team treat this task as a multi-classification, however, we divide those data into 26 categories depending on the first code rather than categorizing directly according to their whole ICD10 codes. We first utilize word2vec model to translate raw text and standard text into real-value vector and chose different dimensions (4,6,10) to train our machine learning model. In the end, models will divide the test data into several kinds ranging from A to Z. The output of machine learning method becomes the input of similarity computation classification.

| ID | FEATURE | DEFINITION | EXAMPLE |
|---|---|---|---|
| 1 | DocID | death certificate ID | 1 |
| 2 | YearCoded | year the death certificate was processed by CépiDC | 2006 |
| 3 | Gender | gender of the deceased | 1/0 |
| 4 | Age | age at the time of death, rounded to the nearest five-year age group | 35 |
| 5 | LocationOfDeath | Location of death | 1 => Home<br>2 => Hospital<br>3 => Private Clinic<br>4 => Hopice, Retirement home<br>5 => Public place<br>6 => Other Location |
| 6 | IntValue | length of time the patient had been suffering from coded cause | if the patient had been experiencing the cause for 6 months, "IntValue" should be 6 and "IntType" should be 4 |
| 7 | CauseRank | Rank of the ICD10 code assigned by coder | 6-1 |
| 8 | StandardText | dictionary entry or exerpt of the raw text that supports the selection of an ICD10 code | surinfection |
| 9 | ICD10 | gold standard ICD10 code | J969 |
| 10 | RawText | The text of 27,850 death certificates | hemorragie digestive |

Table 2: Features of Aligned Data

### 3.3 Similarity computation

We obtain 26 sets according to the algebraic code of the test data using machine learning method and then we apply similarity computation in each set between test data and training data, similarity computation is a method to measure how two words are close to each other in semantic meaning, for example, we get a set S1 in test data which is classified as set A using machine learning method, then we perform similarity computation between set S1 and all text of set A in the training data, we consider the ICD10 code of S1 same as training data where the maximum value obtained.

### 3.4 Combination

We combine our results achieved from regular classification and machine learning classification in order to obtain a perfect performance. We regard the results obtained by regular classification as our baseline. Then, we use the runs achieved by machine learning classification to supply and modify the ICD10 codes in baseline. Suppose S in training data fails to match appropriate text in regular classification, which means that the ICD10 code of S is empty, we treat S as an input of machine learning method and figure out the alphabet code of S. And then, we use dictionaries and standard text in training data to compute the similarity between them. Finally, we chose the most similar text and treat its mapping ICD10 code as the result of S. In this way there is no conflict between the results of regular classification and machine learning classification, since the work of regular rules is based on the classification of machine learning methods.

## 4 Experiments and Evaluation

We utilize the files "AlignedCauses_2006-2012full.csv", "AlignedCauses_2013full.csv", and "AlignedCauses_2014_full.csv" provided by organizers to train and test our methods. We divide the data into training_A set and test_A set according to the ratio of 8 to 2, besides, we also regard the file of 20062013 as training_B set and the file of 2014 as test_B set to validate our approaches. Specifically, we submit two runs based on two methods, where the description for each method is as follows.

Method (A). We utilize machine learning methods first; in which we extract semantic information. In order to avoid the influence of noise information, we use some NLP tools to delete the stop words and do some stemming works. What's more, we manually set some feature sets and find out which set is able to get the most accurate results. And then, we divide training_A into 26 types and each type contains the data with the same alphabet, for example, the beginning of ICD10 codes of data in type A is all A. Finally, we use 26 types data to match regular expressions, compute the similarity, and predict the final results of test_A.

Method (B). We utilize rules mainly based on regular match expressions to extract the ICD10 codes. It means that if the raw text T in test_B set is matched with the raw text R or standard text S in taining_B set, we think the T and R or S have the same ICD10 codes. Because the raw text T in test_B may contain many new written text which haven't appeared in training data, we apply machine learning methods to handle the mismatching data. We select DocID, YearchCoded, Gender, Age, LocationOfDeatch, LineID, RawText, IntType and IntValue as features to train RandomForest and Xgboost model to pre-tag each raw text in test_B. After pre-tagging, we use similarity computation or regular expressions to predict the complete codes.

The primary evaluation measure of this task is the precision, recall and F1. The organizers provide participants with the evaluation program, thus we should use standard program to evaluate our runs.

| FR aligned-ALL | Precision | Recall | F-measure |
|---|---|---|---|
| ECNUica-run1 | 0.7712 | 0.4368 | 0.5577 |
| ECNUica-run2 | 0.7712 | 0.4368 | 0.5577 |
| frequencyBaseline | 0.4517 | 0.4504 | 0.4511 |
| moyenne | 0.7123 | 0.5808 | 0.6343 |
| mediane | 0.7712 | 0.5445 | 0.6407 |

Table 3: The Results of Aligned Data

| FR raw-ALL | Precision | Recall | F-measure |
|---|---|---|---|
| ECNUica-run1 | 0.7895 | 0.4555 | 0.5777 |
| ECNUica-run2 | 0.1 | 0.0 | 0.0001 |
| frequencyBaseline | 0.341 | 0.2005 | 0.2525 |
| moyenne | 0.7228 | 0.4102 | 0.5066 |
| mediane | 0.7981 | 0.475 | 0.579 |

Table 4: The Results Raw Data

## 5 Conclusions and Future Work

In 2018 CLEF eHealth task 1, we propose a regular match expression method and utilize machine learning methods and similarity computation to improve the accuracy of the prediction of the ICD10 codes. However, we still have some problems to solve. The features we select from training data are some normal features, such as age, gender and raw text. In the future, we will pay more attention on the research of extracting useful features and discovering the inner connection of raw data to train machine learning methods.

## 6 Acknowledgement

Suominen, Hanna and Kelly, Liadh and Goeuriot, Lorraine and Kanoulas, Evangelos and Azzopardi, Leif and Spijker, Rene and Li, Dan and Névéol, Aurélie and Ramadier, Lionel and Robert, Aude and Zuccon, Guido and Palotti, Joao. Overview of the CLEF eHealth Evaluation Lab 2018. CLEF 2018 - 8th Conference and Labs of the Evaluation Forum, Lecture Notes in Computer Science (LNCS), Springer, September, 2018. Névéol A, Robert A, Grippo F, Lavergne T, Morgand C, Orsi C, Pelikán L, Ramadier L, Rey G, Zweigenbaum P. CLEF eHealth 2018 Multilingual Information Extraction task Overview: ICD10 Coding of Death Certificates in French, Hungarian and Italian. CLEF 2018 Evaluation Labs and Workshop: Online Working Notes, CEUR-WS, September, 2018.

## References

1. L. Kelly, L. Goeuriot, H. Suominen, A. Névéol, J. Palotti, and G. Zuccon, *Overview of the CLEF eHealth Evaluation Lab 2016.* Springer International Publishing, 2016.
2. L. Goeuriot, L. Kelly, H. Suominen, A. Névéol, A. Robert, E. Kanoulas, R. Spijker, J. Palotti, and G. Zuccon, "Clef 2017 ehealth evaluation lab overview," in *International Conference of the Cross-Language Evaluation Forum for European Languages*, 2017, pp. 291–303.
3. T. Lavergne, A. Névéol, A. Robert, C. Grouin, G. Rey, and P. Zweigenbaum, "A dataset for icd-10 coding of death certificates: Creation and usage," in *Proceedings of the Fifth Workshop on Building and Evaluating Resources for Biomedical Text Mining (BioTxtM2016)*, 2016, pp. 60–69.
4. H. Suominen, L. Kelly, L. Goeuriot, E. Kanoulas, L. Azzopardi, R. Spijker, D. Li, A. Névéol, L. Ramadier, A. Robert, J. Palotti, Jimmy, and G. Zuccon, "Overview of the clef ehealth evaluation lab 2018," in *CLEF 2018 - 8th Conference and Labs of the Evaluation Forum, Lecture Notes in Computer Science (LNCS).* Springer, September 2018.
5. A. Névéol, A. Robert, F. Grippo, C. Morgand, C. Orsi, L. Pelikán, L. Ramadier, G. Rey, and P. Zweigenbaum, "Clef ehealth 2018 multilingual information extraction task overview: Icd10 coding of death certificates in french, hungarian and italian," in *CLEF 2018 Evaluation Labs and Workshop: Online Working Notes.* CEUR-WS, September 2018.
6. M. Pal, "Random forest classifier for remote sensing classification," *International Journal of Remote Sensing*, vol. 26, no. 1, pp. 217–222, 2005.

7. A. Liaw, M. Wiener *et al.*, "Classification and regression by randomforest," *R news*, vol. 2, no. 3, pp. 18–22, 2002.
8. T. Chen, T. He, M. Benesty *et al.*, "Xgboost: extreme gradient boosting," *R package version 0.4-2*, pp. 1–4, 2015.
9. T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*. ACM, 2016, pp. 785–794.