Multi-lingual ICD-10 coding using a hybrid rule-based and supervised classification approach at CLEF eHealth 2017

Jurica Ševa*[1], Madeleine Kittner[1], Roland Roller[2], and Ulf Leser[1]

[1] Humboldt Universität zu Berlin, Knowledge management in Bioinformatics, Berlin, Germany
{seva, kittner, leser}@informatik.hu-berlin.de,
[2] Deutsches Forschungszentrum für Künstliche Intelligenz, Language Technology, Berlin, Germany
roland.roller@dfki.de

Abstract. In this paper we present our research efforts and obtained results within the CLEF eHealth challenge 2017, Track 1. The task involves the recognition and mapping of ICD-10 codes to English and French death certificates. Our approach proposes a two tier, two stage process. First, we use a rule-based system, based on handcrafted rules and the use of Apache Solr, to perform ICD-10 code Named Entity Recognition (NER). This step produces a set of possible candidates extracted from the input text. Next, we use tf-idf weighted character n-gram classification models to normalize and rank a previously generated ICD-10 candidate set. Classification models used are generated and follow the hierarchical structure of the given ICD-10 dictionaries, by creating individual classification models for the first two hierarchical levels (chapters and blocks). Finally, the top candidate, generated from the overlap between the list of possible ICD-10 code candidates (input list) and ranked list of final ICD-10 candidates (output list), is taken as the final ICD-10 code. Although the ICD-10 candidate NER is language-dependent, the normalization and ranking of candidates utilizes a language independent approach.

Keywords: ICD-10 codes, Multilingual Candidates Ranking, Language-independent Information Extraction, Language-independent Information Retrieval, Hierarchical Document Classification, Named Entity Recognition

1 Introduction

In recent years we have witnessed significant advances in automated natural language processing research efforts. This was partly stimulated by the increase of available gold standard corpora as it represents the foundation of scientific research. Research efforts in the field of biomedical text mining (BTM) have been less fortuitous, especially in the domain automatic analysis of electronic health (eHealth) records. This is primarily due to privacy issues and concerns linked with such documents. CLEF eHealth competition [6], through various organized tasks [5, 8], circumvents these restrictions by providing

[*] Corresponding author

gold standard data sets/corpora. Its main focus is on creating automatic information extraction pipelines of valuable information from eHealth documents.

The CLEF eHealth 2017 Task 1[3] [10] serves as an extension of the CLEF eHealth 2016 Task 2 [9]. The goal was to develop a multilingual approach for information extraction of ICD-10 codes from written text. In particular, participants were asked to assign codes from the International Classification of Diseases version 10 (ICD-10)[4] to French and English death certificates. Additionally, it was encouraged to explore multilingual approaches/models as opposed to language dependent models. For both languages customized dictionaries of ICD-10 codes and related annotations were provided by the organizers, not excluding the use of other resources. The task had to be performed fully automatically.

In 2016, the CLEF eHealth ICD-10 coding task was applied to French death certificates only. Participating teams used different rule-based and machine-learning approaches. Ho-Dac et al. [7] for instance used a CRF with various features combined with a rule-based system in order to identify more complex entities. Other participants were using machine-learning approaches such as labeled LDA, SVM, Naive Bayes [4] or treated the task as an information retrieval task using tf-idf models [12]. Van Mulligen et al. [11], the best performing team, extended the terminology with code-term combinations annotated in the training corpus and used a rule-based approach for indexing. Additionally, they processed initial annotations using training data derived precision scores [11].

We approached this years task as a two stage process, by combining NER and document classification to generate the final ICD-10 code. In particular, dictionary-based indexing through Apache Solr[5] was used for Named Entity Recognition and document classification for candidates normalization/ranking. Indexing is based on exact and fuzzy dictionary lookup thus providing potential candidates for a term sequence. The focus of this step was to increase the Recall (R) measure values, by providing a list of potential candidates. Candidates normalization and ranking, through trained classification models, is then applied to rank the list of potential candidates. The focus of this step is the increase of the Precision (P) measure.

Similar to our approach, Zweigenbaum and Lavergne [12] also divided the task into two steps in 2016 to i) generate candidate ICD-10 codes and ii) re-rank candidates. While their approach use tf-idf models for both parts, we use a rule-based system to generate candidate ICD-10 codes. Similarly, the second part of our pipeline models are trained based on the ICD-10 hierarchy, thus include information about dictionary chapters and blocks in our models.

In the following we describe our system and evaluation on training and test data. Compared to all participating systems, our results are well above the average for the French test data, and only average for the English test data.

[3] https://sites.google.com/site/clefehealth2017/task-1

[4] http://www.who.int/classifications/icd/en/

[5] http://lucene.apache.org/solr/

2 Methods

Here we describe the corpora, used terminologies, candidate generation by indexing and candidate ranking using classification.

2.1 Corpora

The French data set is the CépiDC Causes of Death corpus. The corpus contains free text descriptions of causes of death as reported in the standardized causes of death forms. Documents are manually annotated with ICD-10 codes by medical experts. Each document can contain several lines while each line can contain multiple causes and therefore multiple ICD-10 code annotations. Additionally, year of coding, patient age, gender, location of death, and time the patient had been suffering from the coded cause are provided for each document. The English corpus is set up similarly but is provided in a different format. The origin of the data set is not mentioned in the challenge.

Both corpora mostly contain only a few words rather than well-formed sentences, which is common for medical text and a challenge for any NER or Named Entity Normalization (NEN) task. The majority of sentences of death certificates (lines) (about 60%) in the English corpus consist of two to four tokens and two to five tokens in the French corpus. Consequently, as there is almost no context available, the application of machine learning trained models is limited.

The French training set contains 65,843 death certificates from 2006 to 2012 with 264,334 ICD-10 codes annotated. The French test set contains 31,682 documents from 2014 and 2015. The English set is much smaller consisting of 13,329 death certificates from 2015 and 38,908 annotated ICD-10 codes for training and 6,665 documents for testing.

2.2 Terminologies

The organizers provided custom terminologies for both languages. For French six dictionaries are available, related to different years of coding (2006-2015), each providing ICD-10 codes and related terms. Roughly 15% of the terms collected in all dictionaries link to multiple ICD-10 codes with no correlation to the year of coding. Clearly, depending on the context, different ICD-10 codes have been applied. On the other hand, in the provided English terminology each unique term almost always links to a unique ICD-10 code. For supervised classification we used the hierarchy within the ICD-10 terminology as provided here for French and English[6]. The terminology consists of 22 chapters which are divided into blocks and further into classes and subclasses. For instance *Chapter VI: Diseases of the nervous system* contains the block *Inflammatory diseases of the central nervous system* which includes ICD-10 codes G00-G09. The class *G00: Bacterial meningitis, not elsewhere classified* within this block can be further divided into ICD-10 codes like *G00.2: Streptococcal meningitis*. In Section 2.4 we explain how this hierarchy is used to train classifiers for ranking candidate terms.

[6] see `http://www.who.int/classifications/icd/icdonlineversions/en/` and `http://apps.who.int/classifications/icd10/browse/2016/en`

2.3 Candidate generation

To align ICD-10 codes to death certificates, our system applies two methods:

1. ICD-10 code recognition focusing on high R measure values and
2. candidate normalization and ranking to improve P measure values.

ICD-10 candidates are generated, from the input text, based on dictionary look-up and fuzzy search. For both languages, customized dictionaries provided by the organizers are used. Preprocessing of documents and dictionaries has been applied to increase the probability to match the correct concepts. It includes

- conversion to lower case characters;
- removal of punctuation and
- conversion of special characters.

NER follows a stepwise matching strategy. All possible n-grams ($n \leq 5$) of an input text are compared to the dictionary by exact match. If no exact match is found then fuzzy matching is applied using Apache Solr. We allow an edit distance of 1 for each token longer than five characters. Multi-token terms are queued using an AND-query. Solr results are ranked such that the first result contains most of the search tokens while only top 10 Solr results are exported to the candidate list. Overlapping sequences are removed from the candidate list by keeping only the longest matching sequences, which decreases slightly the number of candidates. The resulting list of candidates has a high recall, but a low precision. The following step aims at increasing the precision while keeping a similar level of recall.

2.4 Candidate normalization and ranking

The following step was developed to normalize and rank Section 2.3 output to a single ICD-10 code. For this we used supervised document classification. Unlike the NER process, here we developed a language-independent approach. The following classification models have been taken into consideration while performing model selection and optimization: *Decision Tree Classifier, Random Forest Classifier, Stochastic Gradient Descent Classifier* and *Linear Support Vector Classifier*. Used classification models are based on the content in the first two hierarchical levels of the ICD-10 dictionaries (chapters and blocks) for French and English. Altogether, this pipeline uses 23 different classification models:

1. A single general classification model which classifies the input text to one of 22 ICD-10 chapters and
2. 22 chapter classification models which classify and rank the input text to blocks belonging to the respective ICD-10 chapter.

The normalization and ranking process, as seen in Figure 1, was performed in two stages, representing the (shallow) hierarchical structure of the available ICD-10 dictionaries used to train previously mentioned classification models. The process itself iterates for each input text through the following steps:

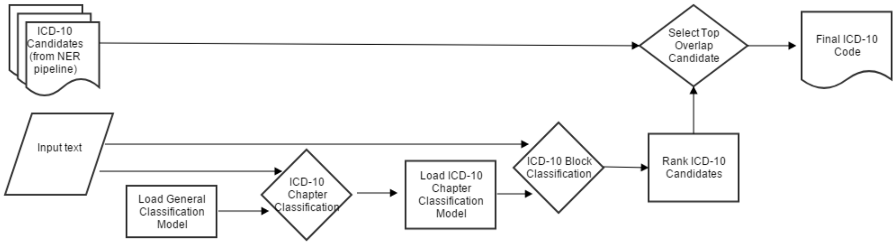
ICD-10 Candidates (from NER pipeline)
Select Top Overlap Candidate
Final ICD-10 Code
Input text
ICD-10 Block Classification
Rank ICD-10 Candidates
Load General Classification Model
ICD-10 Chapter Classification
Load ICD-10 Chapter Classification Model

Fig. 1. Normalization and ranking pipeline for final ICD-10 code selection

1. The input text is assigned to a chapter classification score, $ChapterCS_i$, for each of the 22 ICD-10 chapters, $Chapter_i$;
2. To each block label, $Block_j$, in the respective chapter model, input text is classified and assigned a classification score, $BlockCS_j$, ;
3. A ranking score, RS_x, is calculated as a product of $ChapterCS_i$ and $BlockCS_j$ for each pair ($Chapter_i$, $Block_j$) for each possible ICD-10 candidate label, L_x;
4. A list of ranked ICD-10 codes, LS_{ranked}, is sorted descending by generated ranking score value, RS_x, thus giving us a pair (L_x, RS_x);
5. An overlap list, $LS_{overlap}$, between ICD-10 candidate list, received as output from Section 2.3, and the list of ranked ICD-10 codes, LS_{ranked}, is calculated;
6. Top ranked ICD-10 candidate from $LS_{overlap}$ is selected as the final ICD-10 output code for the input text.

Based on the type of text and the amount of characters available in the training data for each chapter or block labels, character level n-gram features (with *n* between 2 and 5) have been used for building classification models. Extracted features were reinterpreted with tf-idf weighting scheme. This produced a more distinct set of features. Furthermore, tf-idf values were then normalized with L2 norm and feature selection, based on chi^2 test and focusing in top 10% of possible features, was performed. For each of the 23 classification models, model selection and hyper-parameter optimization with randomized search and 10-fold cross validation was performed. This ensured that created models were immune to model overfitting.

Classifier	**#models**
SVM_LinearSVC	13
RandomForestClassifier	6
LogisticRegression	4

Table 1. Frequency of use per optimized classification models

An overview of final models, based on best classification score, and their occurrence number is given in Table 1. Average P, R and F values across all classification models, for the two used hierarchical levels, are given in Table 2.

Level	P	R	F
Chapter	0.880237	0.890911	0.884852
Block	0.920025	0.911876	0.913499

Table 2. Classification models average performance across ICD-10 dictionaries hierarchical levels

3 Results & Discussion

We applied our system to both language sets. Results on the French test set are well above the average results over all participating systems. Test set results on the English data show only average performance. Results for training and test data and performance of individual parts of our system are shown in Table 3. The rule-based NER part referred to as *candidate generation*, and explained in detail in 2.3, focuses on R measure. For the French data sets candidate generation reaches R value of 0.860 for training and 0.844 for test data, while P is low as expected. After *candidate ranking*, explained in 2.4, using the classifier built on the ICD-10 hierarchy R value drops by 0.09 but P value increases to 0.774 for training and 0.800 for test data. For the English data sets we see a similar trend but an overall lower performance. Candidate generation only reaches a R value of 0.76 for training and test sets. Again, after candidate ranking R value drops but here by 0.16, while P value is increased up to 0.61.

Language	Method	Training			Test		
		P	R	F	P	R	F
French	candidate generation	0.548	0.860	0.669	0.557	0.844	0.671
	candidate ranking	0.774	0.770	0.772	0.800	0.751	0.765
	average score				0.648	0.556	0.593
	median score				0.629	0.540	0.548
English	candidate generation	0.305	0.756	0.435	0.320	0.763	0.451
	candidate ranking	0.610	0.610	0.610	0.616	0.606	0.611
	average score				0.655	0.559	0.602
	median score				0.646	0.527	0.589

Table 3. Performance on training and test data for both languages. Performance is given in precision(P), recall(R), and F-measure(F) for each part of our system: after candidate generation and after re-ranking using supervised classification. Only results after candidate re-ranking were submitted. Average and median scores, based on results of all participating teams, are also given.

The different performances for French and English data may be a result of the differences between the datasets. For instance, we did not deal with abbreviations or dissolve coordinated clauses. While they are present in both language sets, we have the impression the English data contains more abbreviations. This could explain the poor performance of the system for the English set. In general, spell checking may improve the overall performance for both systems. Additionally, candidate generation may be improved by taking context information into account.

As far as candidate normalization and ranking is concerned, there are several possibilities how to improve the results. For instance, the current approach, based on optimized language-independent ML models and character level n-grams, ignored other possible features available in the training data (e.g. sex, age, location, etc). Including more diverse data for the classification models would be an interesting next step. One could also look at the entire hierarchical structure of ICD-10. Our ML-models used the first two hierarchical levels of ICD-10 dictionaries. We also tried out a more in-depth classification by creating models below the second level in the ICD-10 dictionary taxonomy. Unfortunately, those approaches failed to produce satisfactory results. This can be attributed to the lack of sufficient data in the supplied training data sets for all possible labels in the taxonomy. Also, we have tested using more complex features like word embeddings which did not yield satisfactory results. This can be explained by the fact that we have used available models not trained on in-domain documents. By using in-language and in-domain documents to produce word embeddings one can expect this approach to be far better. Even though the domain and used language is slightly different and available corpora are small, one could test training word embeddings on available biomedical French and English corpora such as Quaero [3], EMEA [1] or Mantra [2].

Bibliography

[1] Emea corpus. http://opus.lingfil.uu.se/EMEA.php.

[2] Mantra corpus. http://biosemantics.org/index.php/resources/mantra-gsc.

[3] Quaro corpus. https://quaerofrenchmed.limsi.fr/.

[4] Mohammed Dermouche, Vincent Looten, Rémy Flicoteaux, Sylvie Chevret, Julien Velcin, and Namik Taright. ECSTRA-INSERM@ CLEF eHealth2016-task 2: ICD10 code extraction from death certificates. CLEF, 2016.

[5] Lorraine Goeuriot, Liadh Kelly, Hanna Suominen, Leif Hanlen, Aurélie Névéol, Cyril Grouin, João Palotti, and Guido Zuccon. *Overview of the CLEF eHealth Evaluation Lab 2015*, pages 429–443. Springer International Publishing, 2015.

[6] Lorraine Goeuriot, Liadh Kelly, Hanna Suominen, Aurélie Névéol, Aude Robert, Evangelos Kanoulas, Rene Spijker, João Palotti, and Guido Zuccon. CLEF 2017 eHealth Evaluation Lab Overview. *CLEF 2017 - 8th Conference and Labs of the Evaluation Forum, Lecture Notes in Computer Science (LNCS)*, 2017.

[7] Lydia-Mai Ho-Dac, Ludovic Tanguy, Céline Grauby, Nkauj Hnub Aurore Heu Mby, Justine Malosse, Laura Rivière, Amélie Veltz-Mauclair, and Marine Wauquier. LITL at CLEF ehealth2016: recognizing entities in french biomedical documents. In *Working Notes of CLEF 2016 - Conference and Labs of the Evaluation forum, Évora, Portugal, 5-8 September, 2016.*, pages 81–93, 2016.

[8] Liadh Kelly, Lorraine Goeuriot, Hanna Suominen, Aurélie Névéol, João Palotti, and Guido Zuccon. *Overview of the CLEF eHealth Evaluation Lab 2016*, pages 255–266. Springer International Publishing, 2016.

[9] Aurelie Neveol, Lorraine Goeuriot, Liadh Kelly, Kevin Cohen, Cyril Grouin, Thierry Hamon, Thomas Lavergne, Grégoire Rey, Aude Robert, Xavier Tannier, et al. Clinical information extraction at the CLEF eHealth evaluation lab 2016. In *Proceedings of CLEF 2016 Evaluation Labs and Workshop: Online Working Notes. CEUR-WS (September 2016)*, 2016.

[10] Aurélie Névéol, Robert N. Anderson, K. Bretonnel Cohen, Cyril Grouin, Thomas Lavergne, Grégoire Rey, Aude Robert, Claire Rondet, and Pierre Zweigenbaum. CLEF eHealth 2017 Multilingual Information Extraction task overview: ICD10 coding of death certificates in English and French. *CLEF 2017 Evaluation Labs and Workshop: Online Working Notes, CEUR-WS*, 2017.

[11] E Van Mulligen, Zubair Afzal, Saber A Akhondi, Dang Vo, and Jan A Kors. Erasmus MC at CLEF eHealth 2016: Concept recognition and coding in French texts. CLEF, 2016.

[12] Pierre Zweigenbaum and Thomas Lavergne. LIMSI ICD10 coding experiments on CépiDC death certificate statements. CLEF, 2016.