As Simple as Possible: Using the R Tidyverse for Multilingual Information Extraction. IMS Unipd at CLEF eHealth 2020 Task 1

Giorgio Maria Di Nunzio[1,2]

[1] Dept. of Information Engineering – University of Padua
[2] Dept. of Mathematics – University of Padua
giorgiomaria.dinunzio@unipd.it

Abstract. In this paper, we report the results of our participation to the CLEF eHealth 2020 Task on "Multilingual Information Extraction". This task focuses on coding of medical textual data using the International Statistical Classification of Diseases and Related Health Problems (ICD) in Spanish. The main objective of our participation to this task is the study of reproducible experiments that use minimal effort to be set up and run and that can be used as a baseline. The contribution of our experiments to this task can be summarized as follows: the implementation of a reproducible pipeline for text analysis that uses universal dependency parsing; an evaluation of simple classifiers based on perfect matches on different morphological levels together with a tf-idf approach.

1 Introduction

CLEF eHealth is an evaluation challenge in the medical domain where the goal is to provide researchers with datasets, evaluation frameworks, and events. In the CLEF eHealth 2020 edition [1], the organizers set up two tasks to evaluate retrieval systems on different domains. In this paper, we report the results of our participation to the CLEF eHealth Task 1 "Multilingual Information Extraction" [2]. The 2020 task focuses on the evaluation of systems that automatically code clinical textual data in Spanish with ICD codes. In this edition, we continue our line of research that we have been following in the last two years [4, 3]: to study and share reproducible systems that require minimal effort to be run in order to create useful baselines for the research community. In particular, we participated in two of the three subtasks available: subtask 1, ICD10-CM codes assignment to evaluate systems that predict ICD10-CM codes for the classification of diseases; subtask 2 ICD10-PCS codes assignment to evaluate systems that predict ICD10-PCS codes for the classification of medical procedures.

The contribution of our experiments to this task can be summarized as follows:

- the implementation of a reproducible pipeline for text analysis;
- an evaluation of simple classifiers based on perfect matches on different lexical levels and a tf-idf approach.

The remainder of the paper will introduce the methodology and a brief summary of the experimental settings that we used in order to create the runs that we submitted for the task.

2 Method

In this section, we summarize the pipeline for text pre-processing which has been developed in the last two years [4, 3] and has been extended and made reproducible in this work. The source code used in these experiments will be shared online.[3] In general, our method follows the principles described by [?] where the idea is to mine textual information from large text collections in an efficient and effective by means of organized workflows named pipelines. Pipelines are an effective way to manage the sequential process of text analysis by splitting the source code into steps, where the output of one step is the input for the subsequent step. The R programming language has an interesting set of packages that follow this idea, named tidyverse, [4] that we will use in our experiments.

Apart from being a tidy way of organizing software, an important advantage in working with pipelines is that this practice promotes shareability and reproducibility in research workflows which is one of the main pillars in the European Open Science Cloud (EOSC). [5]

2.1 Pipeline for Data Cleaning

In order to produce a dataset ready for training a classifier, we followed the same pipeline for data ingestion and preparation for all the experiments. Instead of using the *tidytext* approach,[6] in this edition we tried the Universal Dependency Parser implementation in R, *udpipe*, which automatically tokenizes, lemmatizes and annotate text.[7]

The following code summarizes all these steps:

```
udpipe_annotate(object = udmodel_spanish,
                x = text,
                doc_id = doc_id_x)
```

where udmodel_spanish is the dependency parser for Spanish, *text* and *doc_id_x* are the textual data and the identifier of each medical document in the dataset. The idea of our approach is to transform each piece of text in order to have

[3] https://github.com/gmdn
[4] https://www.tidyverse.org
[5] https://www.eosc-portal.eu
[6] https://www.tidytextmining.com
[7] https://bnosac.github.io/udpipe/en/index.html

three versions of it: the original tokenized version, the variant with all words lemmatized, the variant with all words stemmed. The following lines take the output of the udpipe step, *annotated_train*, and add the stem version of each token (and transform all text to lowercase):

```
annotated_train %>%
  mutate(stem = wordStem(token, language = ”spanish”)) %>%
  mutate(token_lower = tolower(token)) %>%
  mutate(lemma_lower = tolower(lemma)) %>%
  mutate(stem_lower = tolower(stem))
```

where the %>% symbol represents the usual “pipe” symbol (the output of a function step is the input of the next function), and we used the Spanish Snowball stemmer.

2.2 Classification

The main idea of our simple classifier is based on a memory-based approach with an additional tf-idf weighting scheme. There is no difference between the two subtasks since the procedure is exactly the same:

- choose the morphological level: token, lemma, stem;
- given a sentence that has to be classified, search for any previously classified document that contains that sentence;
- add the classification label to the list of candidates;
- assign the label with the majority of counts.

Since this approach can, in principle, assign only labels that have already been assigned in the past, we added two more steps to include more labels:

- choose the morphological level: token, lemma, stem
- given a sentence that has to be classified, search for any ICD-10 codes that contains the sentence;
- add the classification label to the list of candidates;
- additionally, use a tf-idf to weigh the importance of each word in the sentence;
- assign the label with the largest weight.

3 Experiments

In this section, we briefly describe the setting of official runs that we submitted for this task and the preliminary results sent by the organizers before the workshop.

Table 1. Summary of the results for the two subtasks: upper part subtask 1, lower part subtask 2.

file	MAP	P	R	F1
test_D_only_token	**0.449**	0.373	0.652	**0.474**
test_D_only_token_lemma_stem	0.391	0.306	0.672	0.420
test_D_only_token_lemma_stem_codiesp	0.389	0.299	0.682	0.416
test_D_tfidf_only_token_lemma_stem_codiesp	0.395	0.079	0.699	0.143
test_D_tfidf_only_token_lemma_stem_tfidf_codiesp	0.392	0.081	0.709	0.145
test_P_only_token	0.365	0.310	0.478	**0.376**
test_P_only_token_lemma_stem	0.365	0.291	0.509	0.370
test_P_only_token_lemma_stem_codiesp	0.365	0.291	0.509	0.370
test_P_tfidf_only_token_lemma_stem_codiesp	**0.391**	0.026	0.749	0.051
test_P_tfidf_only_token_lemma_stem_tfidf_codiesp	0.390	0.026	0.747	0.051

3.1 Run Settings

The goal of our experiments is to compare the effectiveness of adding elements to the classifier and study the difference among them in a failure analysis (post-hoc analysis).

We submitted five official runs for each subtask. The letter 'X' in the following description of the run can be substituted with either 'D' or 'P' according to the subtask (Disease or Procedure):

- test_X_only_token: this run uses only a memory-based approach with tokens (original words);
- test_X_only_token_lemma_stem: this run uses only a memory-based approach with tokens, lemmas and stems;
- test_X_only_token_lemma_stem_codiesp: the same as the previous one but we add the description of the ICD-10 codes to the list of possible documents to match
- test_X_tfidf_only_token_lemma_stem_codiesp: the same as the previous one, but we add the tf-idf weights for the token, lemma and stems representation;
- test_X_tfidf_only_token_lemma_stem_tfidf_codiesp: the same as the previous one, but we add the tf-idf weights also for the token, lemma and stems representation of the ICD-10 description.

3.2 Results

A summary of the results for the two subtasks is shown in Table 1. The performance achieved by the combination of elements changes significantly in both subtasks. In general, the simplest classifier that uses only token achieves on average the best performances across different measures. By adding elements to the classifiers, such as lemmas, stems and tf-idf weighting, recall increases at the expenses of precision.

The important decrease of precision when tf-idf is used suggested an additional investigation. In fact, we found a bug in the code that did not activate a threshold on the number of labels retrieved. All the source code will be made available online.[8]

4 Final Remarks and Future Work

The aim of our participation to the CLEF eHealth Task 1 was to test the effectiveness of a simple textual pipeline implemented in R with the 'tidyverse' approach for the problem of classification of clinical textual data. In this task, participants are required to label with ICD-10 codes related to treatment and procedures of health-related documents with the focus on the Spanish language. We tackled this task by focusing on reproducibility aspects, as we did the previous years; this time, we tried a variation of our approach moving from a frequency based classification approach [3, 4] to a sort of memory-based classification by finding perfect matches of previously based classified clinical notes using different lexical variants. This variation was inspired by the baseline produced by organizers of the CLEF 2018 eHealth task [?]. In addition, we included a tf-idf approach to analyze whether the inverse document frequency can help in the classification task.

At the time of writing, we do not have a way to compare our results with those of the other participants, and the comparison with previous years would be useless since the collection of documents is completely different. However, in the preliminary analysis, we found that the token based classification achieved the best results both in terms of classification (F1) and retrieval (MAP) for the disease classification subtask. It was interesting to see that the mixed approach with tf-idf weights performed better in terms of retrieval (MAP) in the procedure classification subtask despite a very low classification score due to an extremely low precision. A preliminary failure analysis showed that the code had a bug that did not allow to weigh and select correctly the labels for the tf-idf approach.

5 Acknowledgements

This work was partially supported by the ExaMode Project, as a part of the European Union Horizon 2020 Program under Grant 825292.

References

1. Lorraine Goeuriot, Hanna Suominen, Liadh Kelly, Antonio Miranda-Escalada, Martin Krallinger, Zhengyang Liu, Gabriella Pasi, Gabriela Saez Gonzales, Marco Viviani, and Chenchen Xu. Overview of the CLEF eHealth evaluation lab 2020. In Avi Arampatzis, Evangelos Kanoulas, Theodora Tsikrika, Stefanos Vrochidis, Hideo Joho, Christina Lioma, Carsten Eickhoff, Aurélie Névéol, and Linda Cappellato and-Nicola Ferro, editors, *Experimental IR Meets Multilinguality, Multimodality, and Interaction: Proceedings of the Eleventh International Conference of the CLEF Association (CLEF 2020)* , LNCS Volume number: 12260, 2020.

[8] `https://github.com/gmdn`

2. Antonio Miranda-Escalada, Aitor Gonzalez-Agirre, Jordi Armengol-Estapé, and Martin Krallinger. Overview of automatic clinical coding: annotations, guidelines, and solutions for non-english clinical cases at codiesp track of CLEF eHealth 2020. In *Working Notes of Conference and Labs of the Evaluation (CLEF) Forum*, CEUR Workshop Proceedings, 2020.
3. Giorgio Maria Di Nunzio. Classification of ICD10 codes with no resources but reproducible code. IMS unipd at CLEF ehealth task 1. In *Working Notes of CLEF 2018 - Conference and Labs of the Evaluation Forum, Avignon, France, September 10-14, 2018.*, 2018.
4. Giorgio Maria Di Nunzio. Classification of animal experiments: A reproducible study. IMS unipd at CLEF ehealth task 1. In Linda Cappellato, Nicola Ferro, David E. Losada, and Henning Müller, editors, *Working Notes of CLEF 2019 - Conference and Labs of the Evaluation Forum, Lugano, Switzerland, September 9-12, 2019*, volume 2380 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2019.