Classification of Animal Experiments: A Reproducible Study. IMS Unipd at CLEF eHealth Task 1

Giorgio Maria Di Nunzio[1,2]

[1] Department of Information Engineering
[2] Department of Mathematics
University of Padua, Italy
giorgiomaria.dinunzio@unipd.it

Abstract. In this paper, we describe the third participation of the Information Management Systems (IMS) group at CLEF eHealth 2019 Task 1. In this task, participants are required to label with ICD-10 codes health-related documents with the focus on the German language and on non-technical summaries (NTPs) of animal experiments. We tackled this task by focusing on reproducibility aspects, as we did the previous years. This time, we tried three different probabilistic Naïve Bayes classifiers that use different hypothesis on the distribution of terms in the documents and the collection. The experimental evaluation showed a significantly different behavior of the classifiers during the training phase and the test phase. We are currently investigating possible sources of biases introduced in the training phase as well as out-of-vocabulary issues and change in the terminology from the training set to the test set.

1 Introduction

In this paper, we report the experimental results of the participation of the IMS group to the CLEF eHealth Lab [3], in particular to Task 1: "Multilingual Information Extraction - Semantic Indexing of animal experiments summaries" [1]. This task consists in automatically labelling with ICD-10 codes health-related documents with the focus on the German language and on non-technical summaries (NTPs) of animal experiments with the German Classification Diseases (ICD10) codes.

The main goal of our participation to the task this year was to test the effectiveness of three simple Naïve Bayes (NB) classifiers and provide the source code (as we did in the previous years) to promote failure analysis and comparison of results.[3]

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

[3] https://github.com/gmdn/CLEF2019

– A study of a reproducibility framework to explain each step of the pipeline from raw data to cleaned data;
– An evaluation of three simple classifiers that use an optimization approach based on the two-dimensional representation of probabilistic models [4, 5].

We submitted 3 official runs, one for each classifier, and we will prepare a number of additional non-official runs that we will evaluate and compare in order to study the change in performance when adding more information in the pipeline.

2 Method

In this section, we summarize the pipeline we used in [6] that has been reproduced in this work for each run.

2.1 Pipeline for Data Cleaning

In order to produce a clean dataset, we followed the same pipeline for data ingestion and preparation for all the experiments. We used the *tidytext* [7] and *SnowballC*[4] packages in R to read and stem words. The following code summarizes these steps:

```
unnest_tokens(term, text,
              token = "words",
              strip_numeric = TRUE) %>%
mutate(term = wordStem(term, language = "de")) %>%
filter(!(term %in% c(stopwords_german, "tier")))
```

The %>% symbol represents the usual "pipe" symbol (the output of a function step is the input of the next function). We used the "unnest_tokens" function to split each text into words; then we stemmed each word with the German Snowball stemmer and filter out the list of stopwords provided by Jaques Savoy.[5] We added the word "tier" ("animal" in German) to the list of stopwords since it is the most frequent word in the collection. We did not perform any acronym expansion/reduction, as we did in the previous years, and we did not perform any word decompounding.

2.2 Classification

We used three NB classifiers for the classification of documents. In particular, the three classifiers differ in the model (the mathematical description) of the distribution of documents and terms. We followed our previous work on the visualization of classifiers for hyper-parameters optimization [2]. The three models are: Multivariate Bernoulli model, Multinomial model, and Poisson model.

[4] https://cran.r-project.org/web/packages/SnowballC/index.html
[5] http://members.unine.ch/jacques.savoy/clef/

In the multivariate Bernoulli model, an object is a binary vector over the space of features:

$$f_k \sim \text{Bern}(\theta_{f_k|c}) \ . \tag{1}$$

where $\theta_{f_k|c}$ is the parameter of the Bernoulli variable of the k-th feature in the c class.

In the multinomial model we have one multinomial random variable which can take values over the set of features:

$$o_j \equiv (N_{1,j}, ..., N_{m,j}) \sim Multinomial(\theta_{f|c}) \ . \tag{2}$$

where $N_{k,j}$ indicates the number of times feature f_k appears in the object o_j.

In the Poisson model, an object o_j is generated by a multivariate Poisson random variable:

$$N_{i,j} \sim \text{Pois}(\theta_{f_i|c}) \ . \tag{3}$$

3 Experiments and Results

We submitted three official runs, one for each model. The goal of these experiments is to compare the effectiveness of the three classifiers and study the difference among them in a failure analysis (post experiments).

3.1 Dataset

The dataset contains 8,793 documents: 7,544 documents for training, 842 for development, and 407 for testing. After we processed the training and development set, the number of features (words) after stemming and stopwords removal is 74,002. There are a total of 233 categories in the German Classification Diseases (ICD10) codes database. The training set contains 230 categories (categories H65-H75, R10-R19, and R20-R23 are missing), the development set contains 156 categories, while the test set 119 categories. Therefore, there are 112 categories that are in the training set but not in the test set and, surprisingly, one category, R10-R19, which is in the test set but not in the training set. Given this distribution of categories, we merged the training and the development set into one dataset that we used to train the classifiers with a k-fold cross validation.

3.2 Evaluation Measures

In order to optimize the three models, we used the F1 measure for each binary classifier (one for each category). In this paper, we report the three measures used by the organizers, Recall, Precision and F1, both the macro-averaged measures (values averaged across all the categories) and the micro-averaged measures (as the sum of all the confusion matrices produced by each classifier). In the tables of the results, we use capital letters to indicate macro-averages (for example Recall) and small letters for micro-averages (for example recall). We report two macro-averaged F1 measures: one computed on the values of the macro-averaged

Table 1. K-fold cross validation Macro- and micro-averaged results

	Pre	Rec	F1	F1*	pre	rec	f1
bernoulli	0.247	0.786	0.375	0.271	0.135	0.628	0.223
multinomial	0.204	0.618	0.307	0.204	0.171	0.642	0.270
poisson	0.727	0.542	0.621	0.468	0.393	0.726	0.510

Table 2. Development Macro- and micro-averaged results (w/out optimization)

optimized	Pre	Rec	F1	F1*	pre	rec	f1
bernoulli	0.275	0.666	0.389	0.281	0.111	0.496	0.181
multinomial	0.200	0.546	0.293	0.206	0.125	0.520	0.202
poisson	0.764	0.609	0.678	0.543	0.408	0.815	0.418
non-optimized							
bernoulli	0.220	0.723	0.337	0.258	0.777	0.201	0.320
multinomial	0.191	0.454	0.269	0.175	0.480	0.365	0.415
poisson	0.764	0.609	0.678	0.543	0.408	0.815	0.418

Precision and Recall, the other (indicated with a '*' at superscript) computed as the average of the F1 measures. For those categories without positive documents (in the development or test set), by default we assign a recall of 1 and a precision of either 0 (when there is at least one false positive) or 1 (when no false positive is found for that category).

3.3 Official Runs

We used a k-fold cross validation approach to train the models and optimize the hyper-parameters of the two-dimensional approach (more details in the final version). We used all the training and development documents for the cross validation with $k = 10$, and we trained a binary classifier for each class in the corpus. Nine out of the ten folds have 838 documents while the latter has 844 documents; consequently, we have on average about 7,540 training documents and 840 validation documents.

The average results across the 233 classes on the ten validation folds are the ones shown in Table 1.

In Table 2, we show the results of the classifiers that use the training set to estimate the probabilities and the development set for the evaluation (with or without the optimization of the decision line). We can see that these results are in line with the one reported during the k-fold cross validation approach with a slightly difference in the micro-averaged performance when the non-optimized version is used compared to the optimized one. This may indicate that there is some overfitting for those categories with too few training documents; in fact, even with just one positive training documents the algorithm tries to find the best fitting line.

Table 3. Test Macro- and micro-averaged results

	Pre	Rec	Fscore	Fscore*	pre	rec	f1
bernoulli	0.530	0.578	0.553	0.226	0.010	0.001	0.001
multinomial	0.492	0.815	0.418	0.614	0.503	0.009	0.017
poisson	0.800	0.566	0.662	0.550	0.039	0.038	0.032

In Table 3, we report the results on the test set. Surprisingly, the behavior of the classifier is completely different from the one we observed during the training/development phase. Macro-averaged measures are still satisfactory, but we have to remind that we introduced a correction in the computation of the recall for those categories without positive documents that may have affected (positively) the averages.

4 Aftermath

We are currently analyzing possible sources of error in the test phase. One thing that seems evident from the analysis is that the distribution of probabilities of terms has changed from the development to the test set. We can observe this from a comparison of Figures 1 and 2. These figures show the two-dimensional distribution ([5]) of positive and negative documents of the Poisson model for development set of category "II". The blue line indicates the decision taken by the classifier: below the line a document is assigned to category II, above the line the document is rejected. The red dots represent the positive documents. We can see that, in this case, the classifier performs well on the development set (a recall of 0.96 and a precision of 0.86) since almost all the positive documents are below the line. On the other hand, the same classifier performs very poorly on the test set (recall and precision both are zero). This is somewhat surprising since in both cases the development and test set contain unseen documents.

We are also studying whether this significant change in the position of the cloud of documents (corresponding to the probability of documents) is related to a different distribution of words in the test set or to a change in the vocabulary of terms in the test set.

There is also a chance in some bug in the source code that we were not able to find until now.

5 Conclusions

In this work, we presented our participation to the CLEF eHealth Task 1 on the classification of medical documents. We presented the evaluation of three probabilistic classifiers based on different assumptions on the distribution of words, namely binary, multinomial and Poisson. We described a method to process the

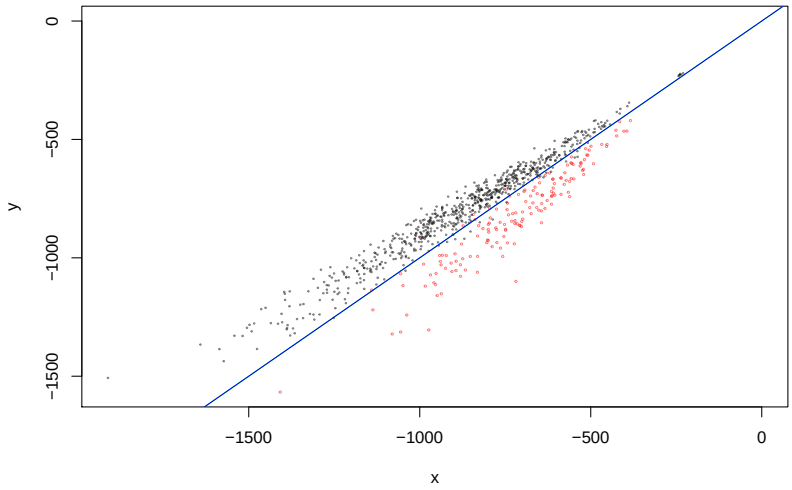

Fig. 1. Poisson model development set. Distribution of positive (red) and negative (black) documents for the category II. The blue line indicates the

documents and optimize the classifiers according to the two-dimensional representation of probabilistic models. The results on the test set were very low despite a correct training/development phase that showed promising results. This opened new ideas about how to better control the training/development phase of the classifier and how to study possible sources of errors in the assumptions made during the training.

References

1. Daniel Butzke, Antje Dötendahl, Nora Leich, Barbara Grune, Mariana Neves, and Gilber Schönfelder. Clef ehealth 2019 multilingual information extraction - semantic indexing of animal experiments. In *CLEF 2019 Evaluation Labs and Workshop: Online Working Notes*. CEUR-WS.org, September 2019.
2. Giorgio Maria Di Nunzio and Alessandro Sordoni. Picturing bayesian classifiers: A visual data mining approach to parameters optimization. In Yanchang Zhao Yonghua Cen, editor, *Data Mining Applications with R*, chapter 2. Elsevier, 2012.
3. Liadh Kelly, Hanna Suominen, Lorraine Goeuriot, Mariana Neves, Evangelos Kanoulas, Dan Li, Leif Azzopardi, Rene Spijker, Guido Zuccon, Jimmy, and Joao Palotti, editors. *Overview of the CLEF eHealth Evaluation Lab 2019. CLEF 2019 - 10th Conference and Labs of the Evaluation Forum*. Lecture Notes in Computer Science (LNCS), Springer, September 2019.
4. Giorgio Maria Di Nunzio. A new decision to take for cost-sensitive naïve bayes classifiers. *Inf. Process. Manage.*, 50(5):653–674, 2014.

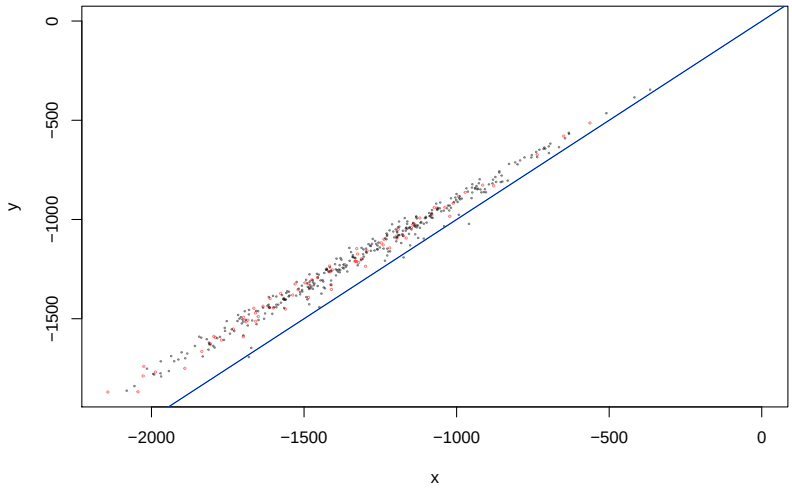

Fig. 2. Poisson model test set. Distribution of positive (red) and negative (black) documents for the category II.

5. Giorgio Maria Di Nunzio. Interactive text categorisation: The geometry of likelihood spaces. *Studies in Computational Intelligence*, 668:13–34, 2017.
6. 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.
7. Julia Silge and David Robinson. tidytext: Text mining and analysis using tidy data principles in r. *The Journal of Open Source Software*, 1(3), 2016.