Informatics in Medicine Unlocked 21 (2020) 100456

Contents lists available at ScienceDirect

Informatics in Medicine Unlocked

journal homepage: http://www.elsevier.com/locate/imu

Underlying cause of death identification from death certificates using reverse coding to text and a NLP based deep learning approach

Vincenzo Della Mea *, Mihai Horia Popescu, Kevin Roitero

University of Udine, Udine, Italy

ARTICLE INFO

Keywords:
ICD-10
Mortality
Automated coding
Machine learning
Deep learning
Embeddings
Natural language processing

ABSTRACT

The identification of the underlying cause of death is a matter of primary importance and one of the most challenging issues in the setting of healthcare policy making. The World Health Organisation provides guidelines for death certificates coding using the ICD-10 classification. Guidelines can be manually applied, but there exist some coding support systems that implement them to simplify the coding work. Nevertheless, there is disparity among countries with respect to the level and the quality of death certificates registration. In this work we propose an effective supervised model based on Natural Language Processing algorithms to the aim of correctly classifying the underlying cause of death from death certificates. In our study we compared tabular representations of the death certificate, including the hierarchical path of each condition in the classification, with a novel representation consisting in translating back to their standard title the conditions expressed as ICD-10 codes. Our experimental evaluation, after training on 10.5 million certificates, reached a 99.03% accuracy, which currently outperforms state-of-the-art systems. For its practical applicability, we studied performance by classification chapter and found that accuracy is low only for chapters including very rare death causes. Finally, to show the robustness of our model, we leverage the model confidence to help identifying death certificates for which a manual coding is needed.

1. Introduction and background

Reliable knowledge on the mortality and causes of death of a population is critical for healthcare policy making. Civil registration and vital statistics systems (CRVS) are the most reliable source of continuous data on fertility, mortality, and causes of death and, if functioning properly, can guide the organisation's policies and priorities for health and development. Cause of death information is one of the most challenging products that comes within the CRVS. Most countries collect information about causes of death by filling death certificates according to a standard methodology defined by the World Health Organisation (WHO) in line with the International Statistical Classification of Diseases and Related Health Problems (ICD). The level of registration of deaths in some countries may be high but there is a huge disparity in generating cause-of-death information across continents and the information on the cause of death is often either absent or of low quality. Overall only around one-third of all the deaths in the world are recorded in civil registries with the associated cause of death information [25]. After being filled by a physician, death certificates are then coded using a specific ICD Revision, currently the 10th Revision (ICD-10). Cause-specific mortality statistics by age and sex are some precious information that can be found in the death certificate, but the so-called underlying cause of death (UCOD) is the most important code used for statistical comparison and public health data. WHO has defined the UCOD as: *"I (a) the disease or injury which initiated the train of morbid events leading directly to death; or (b) the circumstances of the accident or violence which produced the fatal injury."* [26]

The death certificate contains a description of the causal chain of events, which goes from the first health condition that, even remotely, could have caused the death, to the condition that directly brought to death, passing through conditions each one possibly caused by the previous ones. However, normally the last one is not significant from a public health point of view. As example, most chains may be considered ending with a cardiac arrest, but the important condition is the one that initially brought to it - maybe a cancer, an infection, a car accident. The UCOD is in principle one of the conditions of the chain, ideally the first one, but it is not always the case, because chains are reported by the certifying doctor and may include conditions not relevant for the death,

* Corresponding author.
E-mail address: vincenzo.dellamea@uniud.it (V. Della Mea).

https://doi.org/10.1016/j.imu.2020.100456
Received 16 July 2020; Received in revised form 23 September 2020; Accepted 12 October 2020
Available online 16 October 2020

or conditions that, connected together, can be described together as a different condition, etc. Thus, UCOD is not always easy to identify. Furthermore, death certificates are initially filled by the doctor certifying the death with conditions expressed in free text, then coded in ICD-10 by specialised personnel, often under control of a national statistical/epidemiological institution. This aspect also introduces difficulties in the overall process of mortality coding, although it is not among the aims of the present work. For these reasons, many countries adopt a decision support system to help the coder, which is usually different from the certifying doctor.

The WHO provides countries with a standardised format for the definition of the causal chain of events, which is implemented by the set of support systems that are used by most countries. Those systems incorporate a very large number of rules known as "decision tables". Such rules are central to the function of the automated mortality coding systems, but they are also used when performing manual coding; this allows a consistent and harmonised application of the ICD rules across the coding processes.

The decision tables used in the coding process form a knowledge base of relations between pairs of codes representing the causes of death reported on the death certificate that must be taken into consideration during the application of the steps for the selection of the UCOD. This knowledge base was first developed by the US National Center for Health Statistics for the ACME system [14]. Successively it has been embedded in the automated coding system Iris [10] and, since 2011, the Iris Institute maintains the tables according to the WHO official updates of ICD and on the basis of the recommendations of the Mortality Reference Group, which operates in the network of the WHO Collaborating centres for the Family of international Classifications (WHO-FIC). At the present date, Iris and ACME are the most used systems for the support of the automated coding.

An automated coding system for the causes of death reduces considerably the workload of medical coders. When the editing or coding problems can be solved successfully by the support system without manual intervention, the system is expected to handle up to 85% of the death certificates [12]. However, a fair amount of death certificates depict complex situations that cannot be automatically solved and thus are left to manual coders, representing a burden for healthcare systems. Recent studies made in the Netherlands estimated that about 68.5% of death certificates are automatically coded by Iris, leaving 31.5% to manual coders [12]; note that part of them are left to manual coders due to human annotation errors in the certificates, not necessarily to difficulties in UCOD selection.

Machine Learning (ML) is the scientific discipline that focuses on how computers learn from data [19]. ML and in particular Deep Learning (DL) has been employed in the setting of death certificates in at least two tasks: the coding of the free text descriptions to ICD-10, applying/adopting techniques used in the case of discharge letters and patient summaries, and the extraction of UCOD from coded chains of events occurring in death certificates [2,3,9,11,15,27,28].

In particular, the automated UCOD selection using deep learning techniques has been the subject of a couple of papers. Falissard et al. [11] developed a modified Inception network, obtaining an accuracy score of 0.978 on a dataset of 8.5 millions French death certificates, outperforming the Iris performance, which has an accuracy score of 0.925 on automatically coded certificates. In our own previous work [6], a comparison of multiple approaches as Logistic regression, Random Forest, XGBoost and Feedforward Neural Network was presented and state-of-the-art performance is reached using DL techniques. In addition to such techniques, two different encodings were proposed to improve feature organisation and data reduction. More in detail, results from Della Mea et al. [6] obtained an accuracy score of 0.984 (considering the CI, the real accuracy value lies in the [0.984, 0.985] range).

The present paper aims at enhancing the performance obtained in the previous experiment [6], by adding contextual information to the description of the death certificates. This has been carried out following two different approaches: in the former, each code has been enriched with its parents in the classification (ICD-10 has a hierarchical organisation), always as codes. In the latter, categories have been substituted by their descriptive text in order to use Natural Language Processing (NLP) algorithms. In both experiments, the main aim is to exploit the classification hierarchy to provide a better definition of each health condition.

Table 1
Example of a death certificate.

Part 1	*Condition*
1	*I21.9 Acute myocardial infarction*
2	*I10 Hypertension*
3	*N19 Unspecified kidney failure*
4	… … … …. .
5	… … … …. .
Part 2	***Condition***
1	… … … …. .
Other	***Administrative data*** Sex: *female* Age: *55* **Underlying cause of death***I 21.9 Acute myocardial infarction*

2. Data

2.1. The death certificate

The death certificate is the main source of mortality data. Information on the death certificates is best provided by an experienced medical practitioner who is well informed about the medical history of the dead person. This certificate contains administrative details, Frame A which describes the health conditions subdivided in two sections (Part 1 and Part 2) and Frame B which contains additional health conditions (i.e., surgery if performed, autopsy if performed, manner of death, place of occurrence of the external cause, fetal and infant deaths, maternal deaths). The administrative data present in the certificate is used to collect information on sex, date of birth, and date of death of the person. Part 1 normally is formed by 5 lines (but the number could differ between countries), and describes the chain of events which leads to the person death, with the originating cause on the last line and the direct cause on the first line; note that a representation with more than one condition per line is allowed. Part 2 of the certificate is used to annotate the context of the conditions that contributed to the death of the person. Table 1 shows an example of a simplified but realistic death certificate; note that the conditions in the certificates are expressed by both by means of codes and by free text (associated to the code). In this case, for example, according to the rules provided by WHO the UCOD is N40 (prostatic obstruction).

While in principle the UCOD is the originating cause of death and the conditions are the chain of events leading to death, the originating cause of death should be present on the lowest line of the certificate; Nevertheless, from an epidemiological point of view, formal rules are necessary because it is not always the case that such condition (i.e., the lowest line) is the UCOD. For this reason, to ensure that the selection of the UCOD is done in a formal way and following the same principles in every country, the WHO provides a very detailed set of rules to tackle this issue; such rules are defined in the Volume 2 of ICD-10 [26]. This set of rules assure that the data generated in different places is comparable.

2.2. Data source

The death certificates dataset has been obtained from the U.S. National Center for Health Statistics, which make them available for statistical and analytical research.[1] The dataset contains a total of 12, 919,

[1] https://www.cdc.gov/nchs/data_access/vitalstatsonline.htm.

268 records for the years 2014–2017. After a brief pre-processing, the main source of information for the identification of the UCOD has been extracted; we considered the sex, age, and conditions appearing on both Part 1 and Part 2 of the certificate. The train/test split of the data has been made after randomisation and stratified sampling by year; in more detail, for each year 500,000 records have been assigned to the validation set, 100,000 to the test set, and the rest to the training set. Thus, in the end we considered a training set composed by 10, 519, 268 records, a validation set composed by 2,000,000 records, and a test set composed by 400,000 records. We considered a single training/test split as done by Falissard et al. [11].

2.3. Dataset preparation

We considered two different representations, both derived from the second method presented in Ref. [6]. While the first representation is optimized for space reduction and feature relevance, for this work we decided to add additional information for each condition by specifying where it belongs inside the classification. In fact, the ICD classification has a hierarchical structure, and the conditions are coded using leaf codes only. By adding the hierarchical path to the (leaf) condition, we can provide the algorithms with additional information which can be exploited; codes with the same or a similar path will have a similar encoding, resembling the way WHO organises the decision rules used to code the death certificates, and adding the ICD hierarchical structure in the encoding. Such rules generally involve more general terms as sets than ICD-10 leaf codes; more in detail, each set could be seen as a block of ICD codes or also as a group of similar concepts, with a relation between those sets. For this reason in the first encoding we propose in this work we add, for each condition, up to three ICD-10 parents categories. As example, C75 (Malignant neoplasm of other endocrine glands and related structures) has three features associated: Chapter II (C00-D48), Block C00–C97 (Malignant neoplasms) and C00–C75 (Malignant neoplasms, stated or presumed to be primary, of specified sites, except of lymphoid, haematopoietic and related tissue). Conditions appearing in the death certificate are represented by their positions in the certificate, coded with two digits, the most significant for the line, the least significant for the position in the line.

Table 2 shows an example of certificate encoding. We expect this representation to give a boost to the results for the following rationale: Machine/Deep Learning models build a model based on the observations found in the training data, but such category of algorithms treat each observation independently, thus the algorithm cannot integrate into the learning function (at least, not in a naive way) the correlation between codes that might occur for difference instances different instances. For this reason, by coding explicitly the relationship between codes we provide the model a view of the ICD hierarchy, which could lead to possibly increase the model effectiveness.

The second representation we consider consists in substituting ICD codes with their narrative title and then use a NLP algorithm to predict the UCOD. In some way, this reverses the work done by coders because it brings the certificate back to text; however, since titles of conditions that are neighbours in the classification are also most of times similar, this also provides indirect information on the ICD hierarchy. In this representation, each death certificate was encoded in the form of text, where both administrative data and conditions become text. The administrative data was put in an explicit form (e.g., Female, 39y old). Each line of the causality goes between parenthesis, where each ICD-10 code is replaced with the classification title. If multiple codes are on the same line they are concatenated with the expression “or” between the titles (e.g. C16, C80 codes on the same line, are replaced with " (Malignant neoplasm of stomach or Malignant neoplasm, without specification of site)"). Different lines were concatenated with the string “due to”, and the Part 2 codes are connected with the sentence of Part 1 with the string “in the context of”. Three examples can be found in Table 3. The target label was kept as ICD-10 code. The result is a sort of human-readable version of the originally coded text, that attempts to respect the original semantics of the certificate. Tables 2 and 3 show an example of the result of the reverse coding process. After the coding process, we computed some statistic on the dataset. Considering each instance, the average number of words is 20, with a maximum of 186, and a standard deviation of 13.

Note that while many works start from free text to obtain diagnostic codes by means of ML or DL algorithms [4,20,22], to the best of our knowledge this is the first case of application in the medical domain in which we move from a representation based on codes back to a representation based on free text, translating the former into the latter.

3. Methods

3.1. Experiments

The experiments were divided in two main phases: the former is a preliminary one, in which multiple algorithms are taken into account and the training and tests are carried out using a reduced dataset. Then, in the latter experiment we compare and evaluate the best performing algorithms on the complete dataset. Finally, we select the best performing algorithm and we evaluate it on a novel test set composed by the death certificates from the year 2018.

3.2. Machine Learning algorithms

Della Mea et al. [6] have shown a comparison of ML and DL algorithms in the setting of UCOD prediction. Their analysis have shown that the DL algorithms where the most effective ones. For this reason, in this work we consider a set of DL algorithms, namely a Feedforward Neural Network and a set of NLP based deep learning algorithms. The following subsections detail the algorithms considered in this work.

3.2.1. Deep learning and embedding layers for tabular data

We adopted a Feedforward neural network model provided by the Fast. ai framework [13,23]. Our specific architecture model considers 77 features in input, where Sex and Age are taken from administrative data and the remaining 75 features are used for the codes (a pragmatic choice because in our dataset there are no more than 15 ICD-10 codes per certificate, represented with 5 features for each code). After the categorical embeddings encoding, we obtain, depending on the fold, up to 3527 features that are set as input for the first layer, while the output layer has up to 4602 features. The architecture is composed by 5 Hidden layers with 5000 fully connected neurons per layer. Concerning the layer training parameters, we use Rectified Linear Unit (ReLU) as activation function and a Batch Normalisation for continuous variables; the output is computed using the softmax function. We trained the network using 5 epochs, with a maximum learning rate of 2^{-4}.

3.3. Deep learning algorithms for NLP

The algorithms detailed in the next sections use new language representation model with pre-training, which has been shown to be effective for improving many natural language processing tasks, such as sentence-level tasks like natural language inference and paraphrasing [8]. There are two existing strategies for applying pre-trained language representations to so called downstream tasks (i.e., supervised learning tasks that utilise a pre-trained model or component): feature-based and fine-tuning. The feature-based approach uses a task-specific architecture that include the pre-trained representations as additional features. The fine-tuning approach, which we use in this work, introduces minimal task-specific parameters, and is trained on the downstream tasks by simply fine-tuning all pre-trained parameters. We have used many different pre-trained models, detailed in the following.

Table 2
Example encoding for the death certificates as tabular data.

c1	p1	par1.1	par1.2	par1.3	c2	p2	par2.1	par2.2	par2.3	c3 … par 15.3	sex	age	UCOD
C509	10	II	C00–C97	C00–C75	C80	11	II	C00–C97	C76–C80	…	0	39	C509
M726	10	XIII	M60-M79	M70-M79	A419	20	I	A30-A49	A41	…	0	40	M726
C159	10	II	C00–C97	C00–C75	F179	60	V	F10–F19		…	0	37	C159

Table 3
Example encoding for the death certificates as sentence.

Every record of the death certificate as a single sentence	UCOD
Male, 39y old: (Malignant neoplasm of breast, unspecified or Malignant neoplasm, without specification of site)	C509
Male, 40y old: (Sepsis, unspecified) due to (Necrotizing fasciitis)	M726
Male, 37y old: (Malignant neoplasm of oesophagus, unspecified) in the context of (Mental and behavioural disorders … unspecified)	C159

3.3.1. BERT

BERT is a language representation model, which stands for Bidirectional Encoder Representations from Transformers [24]. Unlike other language representation models, BERT is based on the concept of bidirectional training of the transformer attention model to language modelling. The key building block which makes the transformer model particularly effective is the so called encoder/decoder architecture [7, 24]. Such architecture contains different steps: tokenization which consists into splitting each word chunk into pieces; numericalization, which maps each token into a number in the corpus vocabulary; then, a multidimensional embedding is computed for each token; such elements are the ones learned during the training phase. Next, such multidimensional representation is enriched with positional information, in order to modify the representation of a specific word depending on its position in the sentence, in order to capture local context. This sub building block of the Encoder model is called Multi Head (Self) Attention mechanism, which allows to look at other positions in the input sequence (i.e., the context) for indications that can lead to a better encoding for this word. To generate the final output of the transformer based architecture, multiple encoder blocks are chained together in order to capture more abstract and complex representations of the input. Another difference between BERT and its predecessors is the usage of Bidirectional Training: in place of reading the input sequence in one direction, BERT encodes the entire input sequence at once. Furthermore, BERT is trained on two tasks simultaneously: Masked Language Model and Next Sentence Prediction. In the former the model aim to predict a masked word from the input the sequence, in the latter the model is given two input sequences, and it is trained to predict if the second sentence follows the first in a corpus or not.

We considered the BERT-base-uncased[2] which is trained on lowercased English text, BERT-base-cased[3] which is pretrained on the cased version of the same corpus. We also considered other improvements and specialisations of BERT trained on different corpora. BioClinicalBERT[4] is a modification of the BERT model, where the model is initialised with the weights from BioBert, a pre-trained model for biomedical text mining [17], trained on a large-scale biomedical corpora, and trained on all MIMIC notes from the clinical domain [1].

BioBertPubmed[5] is a modification of the BioBert model trained on Pubmed records. Givent the large amount of time and resources needed to train BERT models, researchers proposed a method to pre-train a smaller general purpose language representation model, which can be then fine tuned with good performances on a wide range of tasks like its larger counterparts [21]. This is the case of DistilBERT.[6] While most of the BERT based pre-trained models support more domain specific applications than BERT adding specializations, distilBERT aims at investigating the use of distillation for building task-specific models. With the leverage knowledge distillation during the pre-training phase they have shown that it is possible to reduce the size of a BERT model by 40%, while retaining 97% of its language understanding capabilities and being 60% faster.

All Transformers based models detailed above share a design limitation: the maximum length for textual inputs within such models is set to 512 tokens (i.e., word pieces). To check weather such limitation can arise in our setting, we computed the number of tokens for our dataset considering the BERT-base pre-trained tokenizer (results with other tokenizers are almost identical). We found that considering each instance, the average number of tokens is 40, with a maximum of 275, and a standard deviation of 22.

3.3.2. RoBERTa

RoBERTa,[7] which stand for Robustly optimized BERT approach, propose modifications to the BERT pretraining procedure that improve end-task performance [18]. Specifically, RoBERTa is trained with dynamic masking, full sentences, large mini-batches, and a larger byte-level of BERT [8]. RoBERTa have shown that BERT performances can be substantially improved by training the model longer, with bigger batches and using more data, removing the next sentence prediction objective, training on longer sequences, and dynamically changing the masking pattern applied to the training data. RoBERTa achieved state-of-the-art results on the GLUE, RACE and SQuAD datasets.

3.3.3. XLM

XLM[8] is trained on the same data than the pretrained BERT TensorFlow model [16]. XLM approach show the effectiveness of generative pre-training on cross-lingual pre-training. XLM obtained state-of-the-art results on cross-lingual classification, unsupervised and supervised machine translation tasks. Moreover, XLM improved the previous state-of-the-art for German-English and Romanian-English translations tasks. XLM-R (a variation of XLM), trained on one hundred languages using more than two terabytes of filtered CommonCrawl data, outperformed multilingual BERT (mBERT) on a variety of cross-lingual benchmarks [5].

3.3.4. XLNet

XLNet[9] is an unsupervised language representation learning method based on a novel generalized permutation language modelling objective. Additionally, XLNet employs Transformer-XL as the backbone model, exhibiting excellent performance for language tasks involving long context. Overall, XLNet achieves substantial improvement over previous pre-training objectives on various tasks as: question answering, natural language inference, sentiment analysis, and document ranking [29].

The code used to train the algorithms can be found at https://github.com/MITEL-UNIUD/UCODeep.

[2] https://huggingface.co/bert-base-uncased.
[3] https://huggingface.co/bert-base-cased.
[4] https://huggingface.co/emilyalsentzer/Bio_ClinicalBERT.
[5] https://huggingface.co/monologg/biobert_v1.0_pubmed_pmc.
[6] https://huggingface.co/transformers/model_doc/distilbert.html.
[7] https://github.com/pytorch/fairseq/tree/master/examples/roberta.
[8] https://huggingface.co/transformers/model_doc/xlm.html.
[9] https://huggingface.co/transformers/model_doc/xlnet.html.

4. Results

4.1. Preliminary analysis

On the preliminary exploratory analysis multiple algorithms were selected to be used for a comparison on a smaller version of the dataset; we considered both various NLP algorithms and a FNN. We conducted this preliminary analysis because the main experiment has been proven to be very demanding in terms of resources, as detailed in the respective section. For this analysis we considered a training dataset composed by 400,000 death certificates and a test dataset composed by 100,000 death certificates. We used a set of 800,000 certificates to fine tune the hyperparameters of the FNN. The NLP algorithms have been fine tuned performing 4 epochs of training using the whole training set.

Table 4 shows the effectiveness scores of the algorithms considered as well as the 95% Confidence Interval (CI) for the Accuracy@1 score alone. We computed the CI using the StatsModel library[10]. We consider Accuracy as effectiveness metric, defined as:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN},$$

where: TP represents the true positives (i.e., instances labelled as positive and classified by the algorithm as positive), TN represents the true negatives (i.e., instances labelled as negative and classified by the algorithm as negative), FP represents the false positives (i.e., instances labelled as negative and classified by the algorithm as positive), FN represents the false negatives (i.e., instances labelled as positive and classified by the algorithm as negative). As done in previous work [6,11] we determine the most effective approach according to the accuracy@1 score (i.e., using only the most probable class returned by the algorithm); we report for completeness and future reproducibility also the Accuracy@3 scores (i.e., the accuracy scores computed by considering the top-3 most probable class as returned by the algorithm).

As we can see from Table 4, the FNN is always outperformed by the NLP based models. However, it outperforms the accuracy obtained on the same dataset in our previous work [6], which was 0.936. Concerning the NLP models, although the accuracy scores are very similar, there are some differences. All the BERT based models achieve accuracy@1 scores of around 0.97, with the exception of distilBERT (0.9696) that shown a slightly lower effectiveness score. Overall, we found that the most effective algorithms are: xlnet (accuracy@1 = 0.9746) and xlm (Accuracy@3 = 0.9919). Overall, we see that the NLP-based models obtain very high accuracy scores, at all cut-offs; on the contrary, the FNN architecture lead to obtain significantly lower accuracy. Thus, from the results of this preliminary experiment we can conclude that it appears that the NLP-based representation of death certificates boosts the accuracy levels obtained from leveraging the code-based representation. This effect might be due to the fact that the NLP is able to capture semantic similarities derived from word and sentences gathered from the coding process.

Given the results from the preliminary analysis, to seek confirmation for our findings, we selected a subset of algorithms from Table 4 and we performed the main experiment, detailed in the next section. We remark that this choice, as well as the choice of not using a k-fold validation process, is mainly dictated from the efficiency of the algorithms, and from the training time. To give some statistics, the base BERT models takes around 7 days to train and 2 days to test on a machine with a nVidia Titan XP GPU and 80 GB of RAM memory.

Table 4
Effectiveness scores for the preliminary analysis. We considered 400 k instances in training and 100 k in test.

Method	*Accuracy@1*	95% CI	*Accuracy@3*
FNN fastai	.9480	[.947, .949]	.9816
BERT base uncased	.9705	[.969, .972]	.9842
BERT base cased	.9705	[.969, .972]	.9840
Bio_ClinicalBERT	.9714	[.970, .972]	.9845
BioBert_Pubmed	.9710	[.970, .972]	.9846
distilBERT	.9696	[.969, .971]	.9841
Roberta	.9710	[.970, .972]	.9858
xlm	.9732	[.972, .974]	.9919
xlnet	.9746	[.974, .976]	.9875

[10] see https://www.statsmodels.org/dev/generated/statsmodels.stats.proportion.proportion_confint.html.

4.2. Main analysis

Table 5 shows the effectiveness scores computed when considering the main analysis, that is considering 10 M instances in training and 400 K in test for a subset of algorithms from Table 4. In more detail, we considered FNN for maintaining the hierarchical based coding in the main experiment even if less promising, BERT base uncased and its cased counterpart because they have shown to increase effectiveness scores with the growth of the training data, BIO_clinicalBERT because it has been the most effective BERT variant pre-trained on the medical domain (see Table 4), and xlnet because was the most effective algorithm in the preliminary experiment.

As we can see from Table 5, it is again the case that all the accuracy scores are very similar. As in the previous experiment, all the NLP based models achieve similar scores with accuracy@1 scores of around 0.99; on the contrary, the FNN model shows lower effectiveness scores (i.e., accuracy@1 of 0.9842), this time with no measurable enhancement against the accuracy found in Ref. [6]. Overall, and contrary to what was found in the preliminary analysis, the most effective algorithm in the main experiment is BIO_clinicalBERT, which shows an accuracy@1 score of 0.9903, followed by BERT base uncased and its cased counterpart. The xlnet algorithm which was the most effect on the preliminary experiment appears to be less effective than the BERT based models.

4.3. Accuracy at the chapter level

In this section we break down the performances of the most effective algorithm (i.e., BIO_clinicalBERT) by investigating the classified instances at the ICD chapter level.

Fig. 2 shows on the x-axis the predicted class, on the y-axis the true class for every instance in the dataset from Section 4.2. The values in each cell of the bottom plot represent the relative frequencies normalised such that each row sums up to 1 to show the classification outcome with a focus on each of the ICD chapters (see Fig. 3).

As we can see from Fig. 2 (and as expected from the results of Table 5), almost every ICD chapter is correctly classified in the vast majority of the cases (i.e., more 90% of the times). We also calculated the F1 score per chapter, as detailed in Table 6. Nevertheless, there are some exceptions (Chapters 7, 18, 22), due to the very low number of

Table 5
Effectiveness scores for the main analysis. We considered 10 M instances in training and 400 K in test.

Method	*Accuracy@1*	95% CI	*Accuracy@3*
FNN fastai	.9842	[.984, .985]	.9958
BERT base uncased	.9901	[.990, .990]	.9959
BERT base cased	.9900	[.990, .990]	.9959
Bio_ClinicalBERT	.9903	[.990, .991]	.9961
xlnet	.9898	[.989, .990]	.9960

UCODs belonging to them (see top plot), which make training inadequate.

As additional overall performance measures independent from prevalence, we also calculated the macro-averaged accuracy and F1 scores, as average of equally weighted accuracy and F1 scores per chapter, obtaining respectively 0.974 and 0.968.

4.4. Model confidence

To further break down the BIO_clinicalBERT model performances, as done by Ref. [11], we investigated the model confidence. To do so, we considered the probabilities returned by the model for each instance and the corresponding predicted class.

Fig. 1 shows the probability distributions for the probabilities as returned by the model for the correctly (blue) and incorrectly (orange) classified instances. The plot on the left shows all the y-axis, while the plot on the right shows a potion of x-axis in order to enhance the frequencies for the incorrectly classified instances. As we can see from Fig. 1, the model shows high confidence (i.e., high probabilities) for the correctly classified instances; this is clear by looking at the blue peak in the right part of the left plot. Concerning incorrectly classified instances, from the plot on the right we see that, despite the orange peak around probabilities close to 1, overall the model shows lower confidence (i.e., probabilities lower than 0.2) for the incorrectly classified instances. If we remove from the dataset the instances for which the model is not confident (i.e., the ones with probabilities lower than 0.2) we would maintain the accuracy@1 measure at the 0.9903 score: we move from 396,120 instances correctly classified out of 400,000 to 395,807 instances correctly classified out of 398,655.

Summarising, this analysis has shown that the model is overall less confident on instances that are incorrectly classified; on the contrary, when model is confident on the classification of an instance, in the vast majority of the cases the instance is then correctly classified. This is an important result: in fact we could supply the human annotators with the classification probability of every instance, and they can focus only on instances for which the model is not confident.

4.5. Reusing the model on a new year

A practical useful case is to reuse a model which is previously trained on a set of past instances to classify a set of death certificates for a novel and not previously seen year. In fact, apart from the very rare change of Revision number (ICD-10 was approved in 1990 for adoption in 1994, ICD-11 in 2019 for adoption in 2022), every year a minor update is released, with few changes in codes, and every year also updates to decision tables are released, to keep the pace with ICD-10 but also to correct mistakes in the rules. In our case, we found 210 codes from 2018 that where not present in our training set.

To this aim, we performed the following experiment: we tested the BIO_clinicalBERT model with about 2.85 million death certificates from the year 2018. Table 7 shows the effectiveness scores. As we can see from the table, it is again the case that the FNN model is outperformed by the BIO_clinicalBERT model. Concerning the BIO_clinicalBERT model, we see that it reaches an accuracy score of 0.9875, which is very high considering the fact that there was no instance from the year 2018 in the training set. This is again an important result: we show that a NLP based model can be effectively reused in the setting of a novel year. We want to remark that this is a sort of lower bound for the effectiveness scores of the algorithm on the new year; the performances of the algorithm can be easily boosted by supplying the novel introduced codes in the training/fine tuning phase.

Table 6
Error Rate, Prevalence, and F1 Scores for each Chapter.

Ch.	Prevalence	Error Rate	Error Rate [6]	F1
1	.025	.013	.038	.978
2	.226	.002	.006	.997
3	.003	.061	.109	.933
4	.044	.009	.029	.981
5	.052	.004	.012	.993
6	.071	.006	.011	.994
7	.000	1	.680	–
8	.000	.000	.354	–
9	.307	.004	.012	.993
10	.098	.006	.017	.989
11	.038	.011	.039	.980
12	.002	.035	.066	.965
13	.005	.046	.096	.948
14	.025	.016	.036	.976
15	.000	.114	.110	.854
16	.004	.021	.043	.962
17	.003	.074	.087	.930
18	.012	.004	.006	.997
20	.082	.009	.014	.986
22	.000	1	.000	–

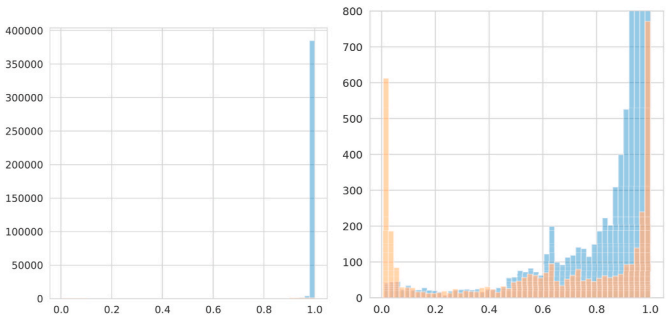

Fig. 1. Probability distributions for correctly (blue) and incorrectly (orange) classified instances. Most effective classifier of Table 5. Full plot on the left, limited y-axis on the right. (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

5. Discussion

From the results detailed in Section 4 we can draw many conclusions: again, our analysis show that the NLP-based representation of death certificates lead to obtain higher accuracy levels than the ones obtained from leveraging the code-based representation, even considering that it has been enriched with the hierarchical structure of ICD with respect to the base version used by Della Mea et al. [6]. These results suggest that it might be the case that the NLP based representation of codes is able to capture semantic similarities derived from word and sentences gathered from the coding process, and that such similarities can not be exploited in the case of a code-based representation.

One limitation of our NLP-based approach is that current models impose limits in the length of text sequences, which in the case of BERT is 512 tokens. However, the realworld death certificates used in our experiments never reached such number, thus this constraint, in our data set, is not influencing results. Larger models have lesser constraints and could be adopted in case of need, although with higher costs in terms of training.

BIO_clinicalBERT is the most effective algorithm on the main experiment (see Section 4.2). This shows that, in the case of UCOD prediction, pre-training the NLP algorithms on a task which is close to the one performed in the fine tuning and test phase lead to obtain slightly higher effectiveness scores. However, the increase from BERT_base is minimal. This can be in part due to the very large training set used in our experiment, which is comparable if not larger than the document set used to fine tune BIO_clinicalBERT. The latter observation seems partly supported by the fact that in the preliminary experiment, with a significantly smaller training set, the difference between the two models is larger.

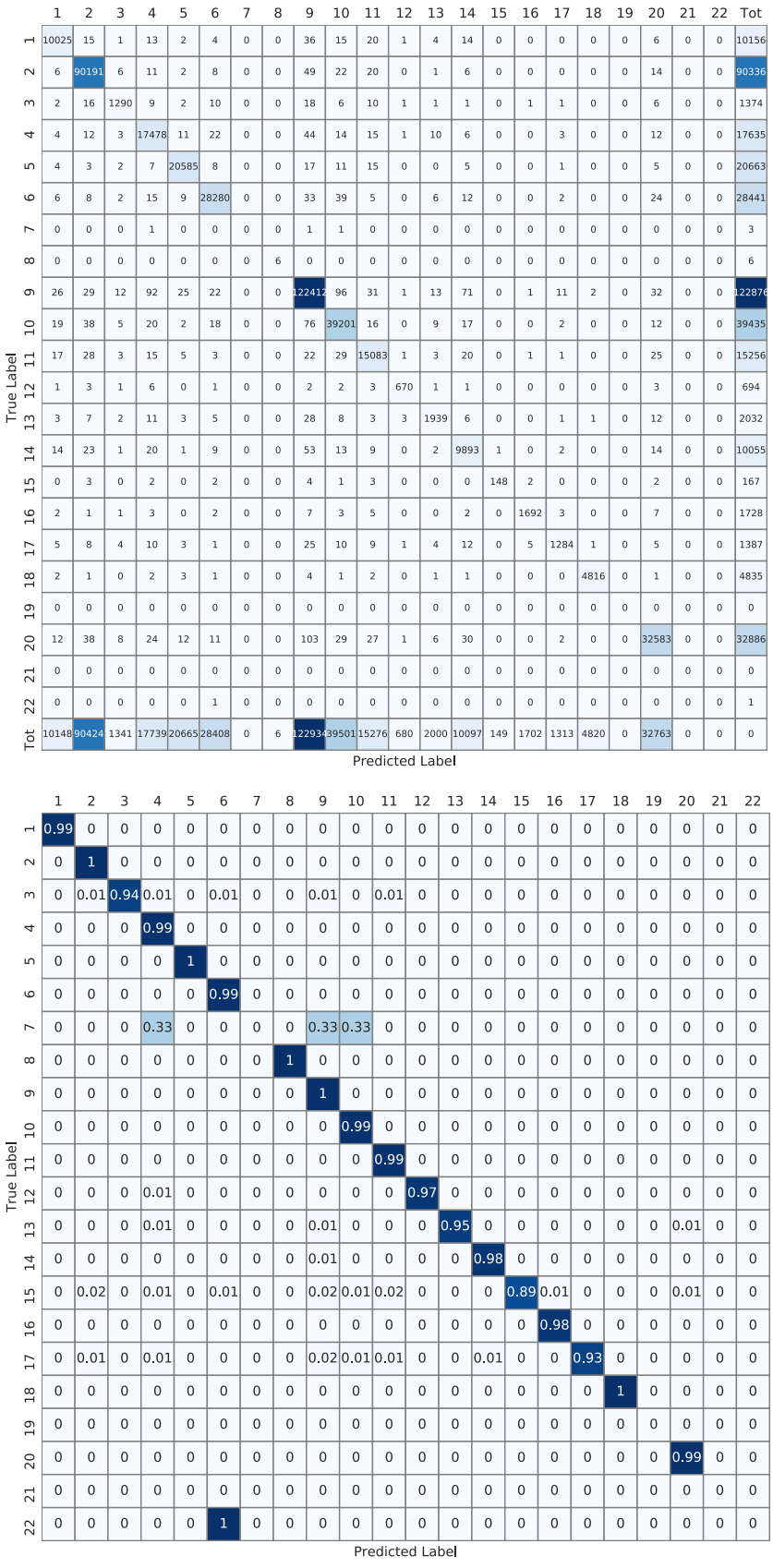

Fig. 2. Accuracy matrix for BIO_clinicalBERT, absolute values (top), and percentages (bottom).

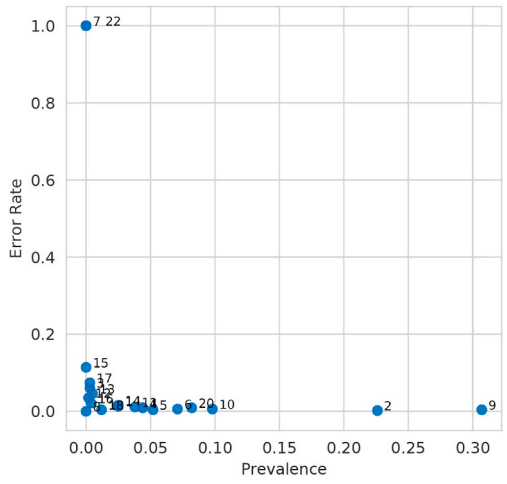

Fig. 3. Error Rate and Prevalence for each Chapter. The exact numbers are detailed in Table 6.

Table 7
Effectiveness scores for the analysis on a novel year. We considered 10 M instances in training (reusing the model from the main experiment without additional training) and 2.85 M in test.

Method	*Accuracy@1*	95% CI	*Accuracy@3*
FNN	.9798	[.980, .980]	.9946
Bio_ClinicalBERT	.9875	[.987, .988]	.9956

Our most performing algorithm shows an accuracy@1 score of 0.9903 (95% CI [0.990, 0.991]). This result outperforms our previous result of 0.9844 (95% CI [0.984, 0.985]), obtained on the same training and test sets [6]. The very first work applying deep learning to UCOD identification, by Falissard et al. [11], reports an effectiveness score of 0.978 (95% CI [0.977, 0.979]) on 8.5 millions French death certificates from 2000 to 2015. However, due to the different data set, not too far in size but created in a different context, with possibly different coding habits, it is impossible to properly compare the effectiveness. What can be told, considering both results, is that deep learning in its various incarnations might provide an effective way for UCOD identification, with the limitation that new codes and rules may need some temporary usage of a rule-based system. Since their work outperforms the state-of the-art software Iris, the same can be said for our model.

Concerning the accuracy at the chapter level (see Section 4.3), we can compare only with our previous results [6], because in Ref. [11] results are presented only visually. Furthermore, chapters 7, 8, 19, 21 and 22, which overall had less than 10 UCOD cases, can be ignored. A chapter-wise comparison with our previous work is shown in Table 6, where error rates are shown side by side together with Prevalence and the F1 scores. Error rate is computed as 1− the Accuracy score, Prevalence details how often each chapter occurs in the dataset, and F1 score is computed as the harmonic mean of Precision and Recall. As we can see by comparing the error rate from our previous work with the from the current work and, excluding very low prevalence chapters, the model presented here always outperforms the previous one. In particular, the error rate in the two most prevalent chapters (2 and 9) is reduced to one third of the previous values. With respect to reusing the model on data from a new year (see Section 4.5), to the best of our knowledge this is the first paper detailing this kind of experiment. However, we concur with Falissard et al. [11] that a rule-based system can be still needed for those chapters scarcely represented in the training set due to their rarity as causes of death.

6. Conclusion

In this work we proposed an effective NLP based model to the aim of identify and correctly classify the underlying cause of death from death certificates, which is applied to disease codes translated back to the text they represent. This result is possibly due to the fact that text re-establishes similarity among conditions, that gets lost with codes used as categories. We have shown that our NLP based approach outperforms both the state-of-the-art and the hierarchically enriched code based representation model. Our results show that our approach reaches an accuracy score of 0.9903 on a test set of 400,000 certificates and an effectiveness score of 0.9875 on a set of previously unseen 2.5 million certificates of a novel year. We show that our approach is robust across the ICD chapters and we found that the model confidence can be leveraged to increase the overall model effectiveness.

A likely, but yet unproven, advantage of the proposed method is the possibility of directly code certificates in their textual form before ICD-10 coding. Among the future works, we want to test this possibility, initially in English to directly exploit our current model, and then with datasets in other languages, which means also a new training after translation of codes to a different language among those in which ICD-10 has been translated.

Another future work is to consider further encodings such as the belonging of a condition to a set of classes used in the rules proposed by WHO and to experiment with a mixture of NLP based and code based representations in order to increase the model performance. We also plan to leverage the model probabilities and to add humans in a human-in-the-loop model in order to mark a step towards an effective model which can be used in practice to support the healthcare decision making and the semi-automatic encoding of death certificates.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

We thank U.S. National Center for Health Statistics for the availability of certificates, Nvidia for providing the GPU used to train the models, and Google for providing Google Cloud credits used to carry out the experiments detailed in this paper.

References

[1] Alsentzer E, Murphy JR, Boag W, Weng WH, Jin D, Naumann T, et al. Publicly available clinical bert embeddings. 2019. arXiv:1904.03323.
[2] Atutxa A, Perez A, Casillas A, Atutxa A, Perez A, Casillas A. Machine learning approaches on diagnostic term encoding with the ICD for clinical documentation. IEEE J Biomed Health Inform 2018;22:1323–9.
[3] Baghdadi Y, Bourrée A, Robert A, Rey G, Gallay A, Zweigenbaum P, Grouin C, Fouillet A. Automatic classification of free-text medical causes from death certificates for reactive mortality surveillance in France. Int J Med Inf 2019;131:103915. https://doi.org/10.1016/j.ijmedinf.2019.06.022. http://www.sciencedirect.com/science/article/pii/S138650561930245X.
[4] Chen P, Barrera A, Rhodes C. Semantic analysis of free text and its application on automatically assigning icd-9-cm codes to patient records. In: 9th IEEE international conference on cognitive Informatics (ICCI'10). IEEE; 2010. p. 68–74.
[5] Conneau A, Khandelwal K, Goyal N, Chaudhary V, Wenzek G, Guzmán F, et al. Unsupervised cross-lingual representation learning at scale. 2019. arXiv:1911.02116.
[6] Della Mea V, Popescu MH, Roitero K. Underlying cause of death identification from death certificates via categorical embeddings and convolutional neural networks. In: ICHI 2020 : IEEE international conference on healthcare Informatics; 2020. In press.
[7] Devlin J, Chang MW, Lee K, Toutanova K. Bert: pre-training of deep bidirectional transformers for language understanding. 2018, 04805. arXiv:1810.
[8] Devlin J, Chang MW, Lee K, Toutanova K. BERT: pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language Technologies, vol. 1. Minneapolis, Minnesota: Association for

Computational Linguistics; 2019. p. 4171–86. https://doi.org/10.18653/v1/N19-1423 (Long and Short Papers), https://www.aclweb.org/anthology/N19-1423.
[9] Duarte F, Martins B, Pinto CS, Silva MJ. Deep neural models for icd-10 coding of death certificates and autopsy reports in free-text. J Biomed Inf 2018;80:64–77. https://doi.org/10.1016/j.jbi.2018.02.011.
[10] Eckert O. [Electronic coding of death certificates]. Bundesgesundheitsblatt - Gesundheitsforsch - Gesundheitsschutz 2019;62:1468–75.
[11] Falissard L, Morgand C, Roussel S, Imbaud C, Ghosn W, Bounebache K, et al. A deep artificial neural network- based model for prediction of underlying cause of death from death certificates: algorithm development and validation. JMIR Medical Informatics 2020;8:e17125.
[12] Harteloh P. The implementation of an automated coding system for cause-of-death statistics. Inf Health Soc Care 2020;45:1–14.
[13] Howard J, Gugger S. Fastai: a layered api for deep learning. Information 11. 2020. https://doi.org/10.3390/info11020108. https://www.mdpi.com/2078-2489/11/2/108.
[14] Israel RA. Automation of mortality data coding and processing in the United States of America. World Health Stat Q 1990;43:259–62.
[15] Koopman B, Zuccon G, Nguyen A, Bergheim A, Grayson N. Automatic icd-10 classification of cancers from free-text death certificates. Int J Med Inf 2015;84:956–65. https://doi.org/10.1016/j.ijmedinf.2015.08.004.
[16] Lample G, Conneau A. Cross-lingual language model pretraining. 2019, 07291. arXiv:1901.
[17] Lee J, Yoon W, Kim S, Kim D, Kim S, So CH, Kang J. Biobert: a pre-trained biomedical language representation model for biomedical text mining. Bioinformatics 2019. https://doi.org/10.1093/bioinformatics/btz682.
[18] Liu Y, Ott M, Goyal N, Du J, Joshi M, Chen D, et al. Roberta: a robustly optimized bert pretraining approach. 2019. arXiv:1907.11692.
[19] Michie D, Spiegelhalter DJ, Taylor C, et al. Machine learning. Neural and Statistical Classification 1994;13:1–298.
[20] Mullenbach J, Wiegreffe S, Duke J, Sun J, Eisenstein J. Explainable prediction of medical codes from clinical text. 2018. arXiv preprint arXiv:1802.05695.
[21] Sanh V, Debut L, Chaumond J, Wolf T. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. 2019. arXiv:1910.01108.
[22] Suominen H, Ginter F, Pyysalo S, Airola A, Pahikkala T, Salanter S, et al. Machine learning to automate the assignment of diagnosis codes to free-text radiology reports: a method description. In: Proceedings of the ICML/UAI/COLT workshop on machine learning for health-care applications; 2008.
[23] fastai team. Categorical embeddings for fastai. https://www.fast.ai/2018/04/29/categorical-embeddings/.
[24] Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, et al. Attention is all you need. In: Advances in neural information processing systems; 2017. p. 5998–6008.
[25] World Health Organization. Civil registration: why counting births and deaths is important. 2014. https://www.who.int/news-room/fact-sheets/detail/civil-registration-why-counting-births-and-deaths-is-important. [Accessed 24 August 2020].
[26] World Health Organization. International statistical classification of diseases and related health problems, 10th revision, vol. 2; 2016. https://icd.who.int/browse10/Content/statichtml/ICD10Volume2_en_2016.pdf. [Accessed 6 July 2020].
[27] Xie P, Xing E. A neural architecture for automated ICD coding. In: Proceedings of the 56th annual meeting of the association for computational linguistics, vol. 1. Melbourne, Australia: Association for Computational Linguistics; 2018. p. 1066–76. https://doi.org/10.18653/v1/P18-1098. Long Papers, https://www.aclweb.org/anthology/P18-1098.
[28] Xu K, Lam M, Pang J, Gao X, Band C, Mathur P, et al. Multimodal machine learning for automated ICD coding. In: Doshi-Velez F, Fackler J, Jung K, Kale DC, Ranganath R, Wallace BC, Wiens J, editors. Proceedings of the machine learning for healthcare conference, MLHC 2019, 9-10 august 2019, ann arbor, Michigan, USA, PMLR; 2019. p. 197–215. http://proceedings.mlr.press/v106/xu19a.html.
[29] Yang Z, Dai Z, Yang Y, Carbonell J, Salakhutdinov R, Le QV. Xlnet: generalized autoregressive pretraining for language understanding. 2019, 08237. arXiv:1906.

Vincenzo Della Mea is associate professor of Medical Informatics and of Advanced Web Technologies at the University of Udine, Italy. He is also head of the Medical Informatics, Telemedicine and eHealth Lab (MITEL). V.Della Mea has been national delegate for the COST Action "EUROTELEPATH" (2008–2011) and local responsible for the EU Marie Curie project "AIDPATH" (2013–2017); he also participated in other national projects. V. Della Mea served as WHO Informatics and Terminologies Committee co-chair for 4 years in the WHO Network of the Family of International Classifications. In the same network he was also member of the Joint Task Force on ICD-11 until the end of mission, and now is member of the ICHI Task Force (International Classification of Health Interventions).

Mihai Horia Popescu is a Research Fellow at the University of Udine, North-East of Italy. His research interests include Natural Language Processing, Computer Vision, Data mining and analysis, Machine Learning, and Statistical Modelling over Medical Domain. He collaborated with multiple organizations in the domain of mortality and morbidity, where he worked on the development of tools supporting automated coding.

Kevin Roitero is a Post Doctoral Research Fellow at the University of Udine, North-East of Italy. His research interests include Information Retrieval Evaluation, Crowdsourcing, Data mining and analysis, Machine Learning, and Statistical Modelling. He visited and collaborated with multiple universities, where he worked on the development of crowdsourcing tasks with the aim of understanding and predicting user features such as user engagement, user agreement, and bias.