UNIVERSITÀ DEGLI STUDI DI UDINE
HIC SUNT FUTURA

DEPARTMENT OF MATHEMATICS, COMPUTER SCIENCE AND PHYSICS

BACHELOR THESIS IN
INTERNET OF THINGS, BIG DATA AND MACHINE LEARNING

Enhancing Medical Code Classification with Ontology-Adapted Contrastive Losses in BERT Models

CANDIDATE
Volpi Davide

SUPERVISOR
Prof. Vincenzo Della Mea

CO-SUPERVISOR
Prof. Kevin Roitero

Academic Year 2023–2024

Institute Contacts
Dipartimento di Scienze Matematiche, Informatiche e Fisiche
Università degli Studi di Udine
Via delle Scienze, 206
33100 Udine — Italia
+39 0432 558400
https://www.dmif.uniud.it/

Acknowledgements

I want to thank all my family who always supported me every day, in every situation, in every good and bad moment, my amazing girlfriend Francesca who always believed in me and encouraged me to do better and better in all the steps of this incredible journey. Thanks goes also to all my friends for all the laughs and happy moments together. The biggest thank goes to Luca, Gianfranco and Devin, my colleagues, but first of all friends, who have spent all this three years with me, between lessons and exams, card games and group projects, jokes and explanations, because without them i would have never been here. Last but not least I want to thank me, because over the years I have never given up in front of hard situations and bad moments and I finally deserved this moment.

Abstract

The use of a standard meaning for textual medical expressions is a fundamental task in epidemiology, statistics and health informatics in order to enable their retrieval, aggregation and interpretation. The main focus of this work is on the death certificates, that are usually coded with the International Classification of Diseases. In this research I employ a BERT based model with an innovative ontology-based approach, which consists of observing the things the data are about and use them as the basis for the data structure, to perform the automatic classification of diagnostic text extracted from death certificates. I show the effectiveness of my proposed work over a set of experiments, where I experiment with many variations of the main algorithm and the related datasets. My results show that using a text coded without a certain string and a Triplet Margin with Distance Loss, with a specific combination of anchor, positive and negative example to form a triplet, outperforms the state of art, reaching an Accuracy of 96.6% on the leaf level classification and of 97.0% at the category level classification.

Contents

List of Figures

List of Tables

1

Introduction

In this first chapter at first, I'm going to make a general overview about the problem and the solution that I'm going to implement (section 1.1). Secondly, in section 1.3 i will introduce the ICD topic and its applications, then in section 1.4 I'm going to present the Large Language Models.

1.1 The experiment

Nowadays, the association of a standardize meaning to textual expressions, in order to enable their retrieval, aggregation and interpretation is a fundamental task for epidemiology, statistics and health informatics [18, 25]. The International Classification of Diseases (ICD) is a standard used, in one of its revisions, for coding death certificates. There are some support tools to help coding the causes of death such as Iris [10] and Acme [14] but studies made in Netherlands estimated that only about 68.5% of death certificates are automatically coded by Iris, leaving the other 31.5% to manual coders [12]. As a result,the automatic classification for the death certificates is a problem that needs improvement in terms of efficiency and accuracy in order to reduce considerably the workload of medical coders. This problem can be solved through to the use of many Artificial Intelligence models.

In this study I'm going to explore new frontiers in the training of Large Language Models (LLMs) by using an innovative ontology-based approach. This method will use ontologies to define and quantify the semantic distance between tree concepts, then use it as the basis in a custom loss function to train a model. Through this approach I'm going to develop new algorithms that not only improve the comprehension and generation of the text from the model, but also his ability to capture and reflect the complex and subtle relationships between concepts.

The goal of this work is to improve the performance of the automatic death certificates classification, using ICD codes, through the use of a Large Language Model (LLM), starting from previous experiments [7, 20], and trying to outcome better results. The model chosen for this experiment is a Bidirectional Encoder Representations from Transformers (BERT) with the use of a Triplet Margin Loss with a Custom Distance ontology-based on the ICD-10 [4] hierarchy.

1.2 Ontology Based Approach

In 1998, Studer et al.[24] stated that "An ontology is a formal, explicit specification of a shared conceptualization". Now an ontology in Computer Science is a description of data structure of classes, properties, and relationships in a domain of knowledge. It is meant to serve as a bias for instances of knowledge graphs, ensuring data consistency and understanding of the data model [19].

The ontology-based approach consists in looking at the things the data is about and use them as the basis for the structure of the data. If you correctly identify the things that are important to the business, and the relationships between them, then you will have developed a data model in 6th Normal Form. The goal is to represent one thing once and control duplication to maximize data quality[28].

1.3 International Classification of Diseases

The ICD is designed by the World Health Organization (WHO) to promote the international comparability in collect, classify and present the mortality statistics through death certificates. This classification has been revised periodically and in this work I'm going to use the Tenth revision (ICD-10), which was endorsed by the World Health Assembly in 1990 and has been in use since 1 January 1993. Initially, it contained 1967 items and now consists of about 12000 codable entities. Now the ICD-10 is adopted by more than 100 WHO member states and most individual nations. However, not every country in the world record the causes of death [22]. In the figure 1.1 in green you can see the mortality with cause of death available to WHO.

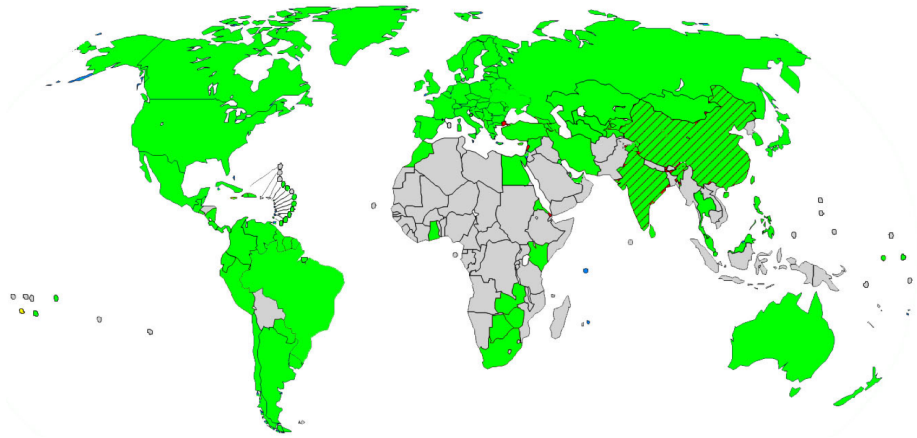

Figure 1.1: Registration of Causes of Death [26]

ICDs in their 10th version are a 3-digit/character base alphanumeric codes with extensions and subcategories, up to 5-6 characters, as shown in the table 1.1. They are organised in a hierarchical structure (Figure 1.2).

Code	Description
A04.7	Enterocolitis due to Clostridium difficile

Table 1.1: Example of ICD-10 code

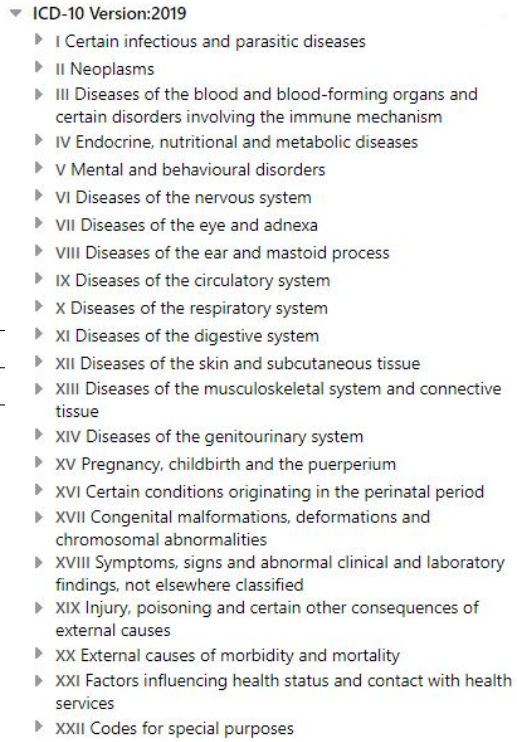

Figure 1.2: Class hierarchy of ICD-10 [13]

Every nation can make modifications to allow for clinical usage. For instance, the ICD-10 clinical modification (ICD-10-CM) is used in the United States and is expanded to about 68000 entities.

The newest revision of ICD is the eleventh (ICD-11), which was approved in 2019 and endorsed in 2022.

1.4 Language Models

Language Models (LMs) are a category of probabilistic models made to identify and learn statistical patterns in natural language. The main goal of a Language Model (LM) is to predict the most probable word or words that succeeds a given input sentence. Firstly, a model need to be pre-trained on generic data and then fine-tuned on specific data as in figure 1.3.

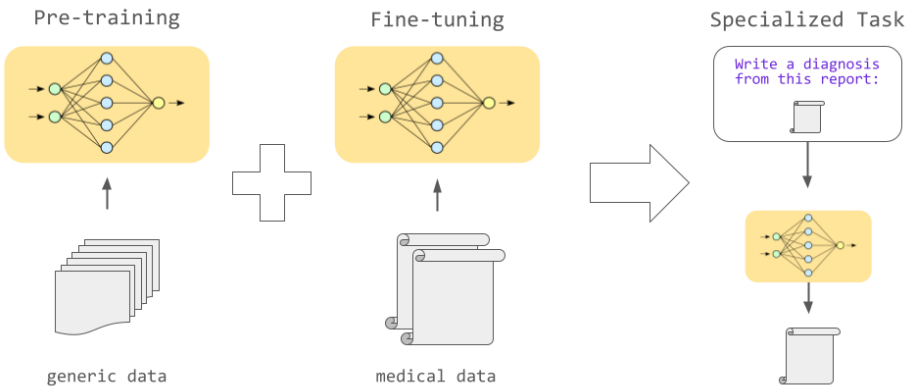

Figure 1.3: Pre-training and Fine-tuning [2]

A LLM is an automated learning model of Artificial Intelligence (AI) that is specialised in understanding and generation of text in Natural Language (NL) in a coherent and contextually relevant way. The main difference between a LM and a LLM is that LLM have more parameters. This leads to achieve better performance but on the other hand requires more computational resources and training data to reach their full potential. LLMs are trained on large datasets with Machine Learning (ML) and Deep Learning techniques (DL). They are built on a neural network called a transformer network. Furthermore, LLMs are very important due to their incredible flexibility in terms of operations and results and there are three main language models:

- **Generative Pre-Trained Transformer (GPT):** specialised in text generation.
- **Bidirectional Encoder Representations from Transformers (BERT):** specialised in text comprehension and classification.
- **Text-to-Text Transfer Transformer (T5):** specialised in automatic translation and text synthesis.

The principal element of LLMs functioning are transformers, invented in 2017. They are very efficient at processing large chunks of data at once through parallelization, this leads to allow to train on bigger datasets than previous architectures. Their extremely good functioning is due to:

- **Word Embeddings:** high dimensional vector representations of words that numerically capture their semantic and syntactic properties. Instead of treating words as isolated entities, word embeddings allow the model to learn and understand the complex interaction of words in a given context.
- **Attention Mechanisms:** allows the model to assess the importance of different words or phrases in the text. This leads the model to target specific parts of the input, assigning different attention scores to words based on their relevance to the task being performed.

All the transformers-based language models use an encoder-decoder architecture to process and generate text:

- **Encoder (figure 1.4):** converts the input text into a numerical, high-dimensional, geometrically and statistically significant representation, normally by processing the word embeddings and incorporating the attention mechanism.
- **Decoder (figure 1.4):** transforms the encoded text into output text by using the attention mechanism to decode it back into text.

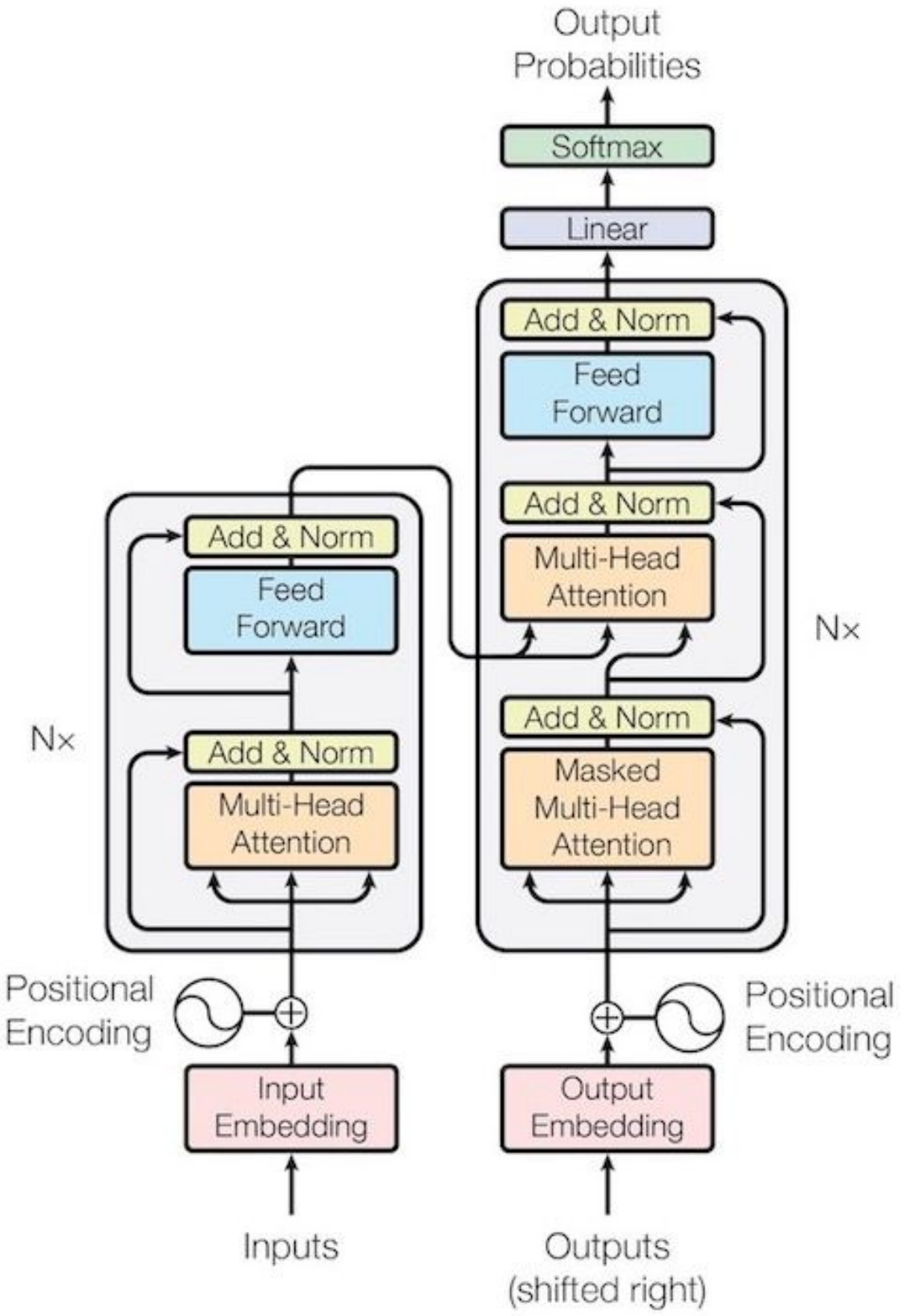

Figure 1.4: The Transformer - model architecture [27]

2

Related work

In this second chapter I'm going to introduce the state-of-art of the automatic ICD classification with the main and latest works (section 2.1,2.2,2.3,2.4, 2.5 and 2.6). Then, I will introduce the state of art of the hierarchical classification and latest works (section 2.7, 2.8 and 2.9).

2.1 Automatic ICD-10 classification of cancers from free-text death certificates

Koopman et al.[15] in 2015 provided automatic identification and characterisation of cancers from large collections of free-text death certificates. This allowed organizations such as Cancer Registers to monitor and report on cancer mortality in a timely and accurate manner. The proposed system had tow components: a natural language processing pipeline that extracts features form death certificates; and a series of supervised Support Vector Machines, that utilise the extracted features for classification. The methods and findings of this study are generally applicable; they can be transferred to other ICD-10 classification task beyond cancer classification and to other source of medical free-text besides death certificates.

2.2 Deep neural models for ICD-10 coding of death certificates and autopsy reports in free-text

Francisco Duarte et al.[9] in 2018 addressed the assignment of ICD-10 codes for causes of death by analyzing free-text descriptions in death certificates, together with the associated autopsy reports and clinical bulletins, from the Portuguese Ministry of Health. They leveraged a deep neural network that combines word embeddings, recurrent units, and neural attention, for the generation of intermediate representations of the textual contents. The neural network also explored the hierarchical nature of the input data, by building representations from the sequences of words within individual fields, which are then combined according to the sequences of fields that compose the inputs. Moreover, they explored innovative mechanisms for initializing the weights of the final nodes of the network, leveraging co-occurrences between classes together with the hierarchical structure of ICD-10.

2.3 A strategy of analysis of free-text E-death certificates using machine learning

Yasmine Baghdadi et al.[3] in 2019 showed that the use of free-text causes of death for reactive mortality surveillance requires the development of a strategy for the analysis of these data. Defining Mortality syndromic groups (MSGs) was essential for the implementation of automatic classification methods of the death certificates. The dynamic of MSGs using ICD-10 codes or Support vector machines classification were comparable. However, the use of ICD-10 codes for reactive mortality surveillance was not an option due to the delay of availability of the codes. The uses of machine learning methods, thus, enable to harness free-text causes of death for the reactive mortality surveillance with an objective of detection and early impact assessment.

2.4 A Deep Artificial Neural Network-Based Model for Prediction of Underlying Cause of Death From Death Certificates: Algorithm Development and Validation

Falissard et al.[11] in 2020 showed that deep artificial neural networks are perfectly suited to the analysis of electronic health records and can learn a complex set of medical rules directly from voluminous datasets, without any explicit prior knowledge. Although not entirely free from mistakes, the derived algorithm constitutes a powerful decision-making tool that is able to handle structured medical data with an unprecedented performance.

2.5 Underlying cause of death identification from death certificates

A previous work in 2020 from Della Mea et al. [7] proposed an effective supervised model based on Natural Language Processing (NLP) algorithms in order to correctly classifying the underlying cause of death from death certificates. In their study they compared tabular representations of the death certificates, which include the hierarchical path of each condition in the classification, with an innovative representation that consist in translating the conditions expressed as ICD-10 codes back to their standard title. Their experiment evaluation, after training on 10.5 million certificates, achieved an accuracy of 99.03%, exceeding the state-of-the-art systems. They also studied performance according to chapter classification and found that accuracy is only low for chapters that include very rare death causes. They also exploited the model confidence to help identify death certificates for which a manual coding is required.

2.6 Automatic assignment of icd-10 codes to diagnostic texts

Then, in 2021, further work from Della Mea et al. [20] proposed both classical ML and BERT based models to perform the automatic classification of diagnostic texts extracted from death certificates. The researchers demonstrated the effectiveness of their proposed approach through a set of experiments, in which they tested different set of features and algorithm variants. The results showed that BERT

based models, and in particular the ones pre-trained on the specific domain, outperform classical ML algorithms, achieving an Accuracy and F1-Score of 0.952 and 0.943, respectively.

2.7 A Review of Performance Evaluation Measures for Hierarchical Classifiers

Costa et al.[6] in 2007 stated that criteria for evaluating the performance of a classifier are an important part in its design. They allow to estimate the behaviour of the generated classifier on unseen data and can be also used to compare its performance against the performance of classifiers generated by other classification algorithms. For hierarchical classification problems, where there are multiple classes which are hierarchically related, the evaluation step is more complex. In this survey they reviewed some of the main evaluation measures for hierarchical classification models. They observed that there was not yet a consensus concerning which evaluation measure should be used in the evaluation of a hierarchical classifier.

2.8 Evaluation Measures for Hierarchical Classification: a unified view and novel approaches

Aris Kosmopoulos et al.[16] in 2013 studied the problem of evaluating the performance of hierarchical classification methods. Specifically, their work abstracted and presented the key points of existing performance measures. They proposed a grouping of the methods into pair-based and set-based. Measures in the former group attempt to match each each prediction to a true class and measure their distance. In contrast set-based measures use the hierarchical relations in order to augment the sets of predicted and true labels, and then use set operations, like symmetric difference and intersection, on the augmented label sets. In order to model pair-based measures, they introduced a novel generic framework based on flow networks, while for set-based measures they provided a framework based on set operations. Another contribution of this paper was the proposal of two measures that address several deficiencies of existing measures. The proposed measures, along with existing ones were assessed in two ways. First, they applied them to selected cases, in order to demonstrate their pros and cons. Second, they studied them empirically on three large datasets based on DMOZ and Wikipedia with different characteristics. The analysis of the results showed that the hierarchical measures behave differently, especially in cases of multi-label data and DAG hierarchies.

2.9 Evaluating Extreme Hierarchical Multi-label Classification

In 2022, Enrique Amigó and Agustín D. Delgado [1] analysed the state of art of evaluation metrics based on a set of formal properties and they defined an information theoretic based metric inspired by the Information Contrast Model (ICM). Their experiments on synthetic data and a case study on real data showed the suitability of the ICM with respect to existing metrics.

3

Methods

In this third chapter, firstly, I will present an overview of the methods (section 3.1). Secondly, I'm going to explain the model and the different loss function I'm going to use, respectively section 3.2, 3.3 and 3.4. Then, I will show some improvements to previous methods in order to achieve better results (section 3.5 and 3.6). Finally, in section 3.7 I'm going to present how the the final dataset variation will be conducted.

3.1 Experiments

The experiment will be divided in two main parts: the first one where the model will be trained and tested on reduced datasets, respectively of 50000 and 5000 instances, with different algorithms. Then, in the second part, all the results will be compared and the best methods will be refined with other techniques. Finally, the best performing algorithm and the base model will be trained and tested on larger datasets, respectively of 500000 and 100000 instances, to observe the final results.

3.2 BERT

The BERT model, also known as Bidirectional Encoder Representations from Transformers [27, 8], use Transformers, as shown in chapter 1.4. It use only the Transformers encoder mechanism to generate a language model. As opposed to directional models, the Transformers encoder reads the entire sequence of words at once. This characteristic allows the model to learn the context of a word based on all of its surroundings.

The BERT model I will use for this work will be a Bert Base Uncased, the smallest pre-trained BERT version, with a tokenize function, a function that splits a string into single tokens based on a regular expression, with max_lenght of 512, batched and with 3 training epochs.

Firstly, the model will be trained on a dataset of 50 thousand certificates and tested on a dataset of 5 thousand certificates. Lastly, the best performing algorithm will be trained on a dataset of 500 thousand certificates and tested on a dataset of 100 thousand certificates.

All computing operations will be done on the GPU due to the higher computing power than the CPU.

3.3 Cross Entropy Loss

Through the entropy you can calculate the grade of disorder inside a system. In information theory, the entropy of a single variable or event measures its level of uncertainty. The entropy H of an event can be calculated using the *Shannon entropy Equation*

$$H(X) = -\sum_{i=1}^{n} p(x_i) \log p(x_i)$$

where the $p(x_i)$ are the odds of all the possible outcomes.

Generally, Cross Entropy is a loss function used in supervised learning and basically, the BERT model use a $CategoricalCrossEntropyLoss$ to measure the performance of his multi-class classification with the formula

$$L = -\frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{M} y_{ij} \log(p_{ij})$$

where N is the number of samples, M is the number of classes, y_{ij} is a binary value of 1 if the classification i belongs to the class j, otherwise 0, and p_{ij} is the predicted probability that the i classification belongs to the class j. It measures the difference between the discovered probability of a classification model and the predicted values. Cross Entropy use this difference to regulate the weights of a ML model during training [17].

3.4 Triplet Margin With Distance Loss

A Triplet Margin Loss aims to minimize the distance between an anchor and a positive example, and maximize the distance between the anchor and a negative example. Usually, this loss function is used in supervised similarity or metric learning but in this work I'm going to rework it and use it for a hierarchical ontological classification.

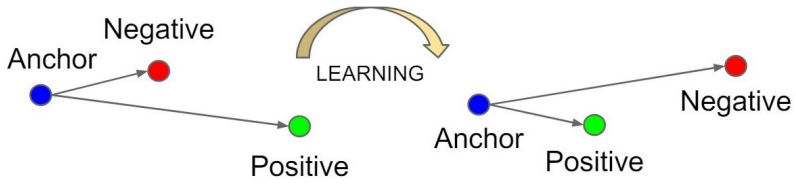

Figure 3.1: Example of Triplet Loss [23]

The mathematical loss value L can be calculated as

$$max(d(a,p) - d(a,n) + m, 0)$$

where p is the positive example, n is the negative example, a is the anchor and m is a margin that represents the minimum difference between the positive and negative distances required for the loss to be 0. Then d is a distance function used to measure the distance between the three samples. By default in a Triplet Margin Loss the function uses $EuclideanDistance$ [23] but through the use of a Triplet Margin With Distance Loss [21] in this work I'm going to use a custom $TreeDistance$ based on the ICD-10 ontology.

This function will be built with the help of the library *simple_icd_10*, a simple python library for ICD-10 codes [5], and will calculate the distance between two input nodes as a classic tree distance algorithm.

The triplet selection will be a crucial point in this work and in the next sections I'm going to explore various different combinations in order to find the best classification algorithm for this case study, based on the neural network inputs.

3.4.1 Version 1

Assuming that the input is:

Code	Text
J18.9	Male, 80y old: Respiratory failure, unspecified due to Pneumonia, unspecified

Table 3.1: Example of input death certificate in Version 1

The first triplet I'm going to implement consists in:

- **Anchor:** The textual description of the input label parent.

Code	Description
J18	Pneumonia, organism unspecified

Table 3.2: Example of anchor in Version 1

- **Positive example:** Another text with the same label of the input.

Code	Text
J18.9	Male, 70y old: Sepsis, unspecified due to Pneumonia, unspecified

Table 3.3: Example of positive example in Version 1

- **Negative example:** Another text with a different label of the input (it must be a leaf-label).

Code	Text
J18.1	Female, 86y old: Acute respiratory failure due to Lobar pneumonia, unspecified

Table 3.4: Example of negative example in Version 1

This approach leads to the closing of the label cluster with the parent cluster and the distancing of another random leaf-cluster.

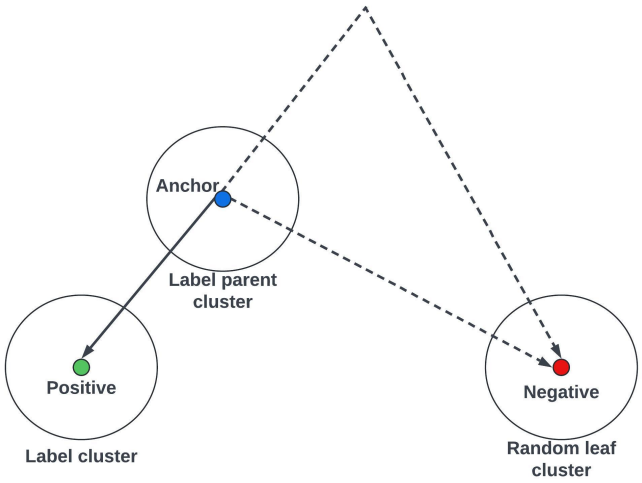

Figure 3.2: Example of Triplet Loss Version 1

3.4.2 Version 2

Assuming that the input is:

Code	Text
J18.9	Male, 80y old: Respiratory failure, unspecified due to Pneumonia, unspecified

Table 3.5: Example of input death certificate in Version 2

The second triplet I'm going to implement consists in:

- **Anchor:** The textual description of the input label parent.

Code	Description
J18	Pneumonia, organism unspecified

Table 3.6: Example of input anchor in Version 2

- **Positive example:** Another text with the same label of the input.

Code	Text
J18.9	Male, 70y old: Sepsis, unspecified due to Pneumonia, unspecified

Table 3.7: Example of positive example in Version 2

- **Negative example:** Another text with a different label from the input (it must be a leaf-label) but not an input label brother.

Code	Text
G11.0	Male, 36y old: Respiratory failure, unspecified due to Congenital nonprogressive ataxia

Table 3.8: Example of negative example in Version 2

This approach leads to the closing of the label cluster with the parent cluster and the distancing of another random leaf-cluster but not a label input brother. That is because, in Version 1 (section 3.4.1), in two different steps of the loss there might be two mirrored pairs of negative and positive examples such as (J18.9 ; J18.1) and (J18.1 ; J18.9). This would result in a cancellation of the two losses due to their reciprocity.

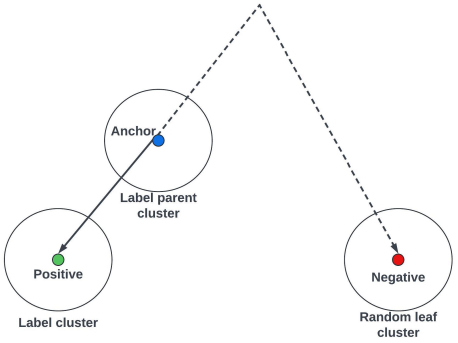

Figure 3.3: Example of Triplet Loss Version 2

3.4.3 Version 3

Assuming that the input is:

Code	Text
J18.9	Male, 80y old: Respiratory failure, unspecified due to Pneumonia, unspecified

Table 3.9: Example of input death certificate in Version 3

The third triplet I'm going to implement consists in:

- **Anchor:** Another text with the same label of the input.

Code	Description
J18.9	Pneumonia, unspecified

Table 3.10: Example of anchor in Version 3

- **Positive example:** The textual description of the input label.

Code	Text
J18.9	Male, 70y old: Sepsis, unspecified due to Pneumonia, unspecified

Table 3.11: Example of positive example in Version 3

- **Negative example:** Another text with a different label from the input (it must be a leaf-label) but not an input label brother.

Code	Text
G11.0	Male, 36y old: Respiratory failure, unspecified due to Congenital nonprogressive ataxia

Table 3.12: Example of negative example in Version 3

Through this approach the model should improve the classification at the leaf levels by bringing closer texts with the same labels and spreading out texts with a different label.

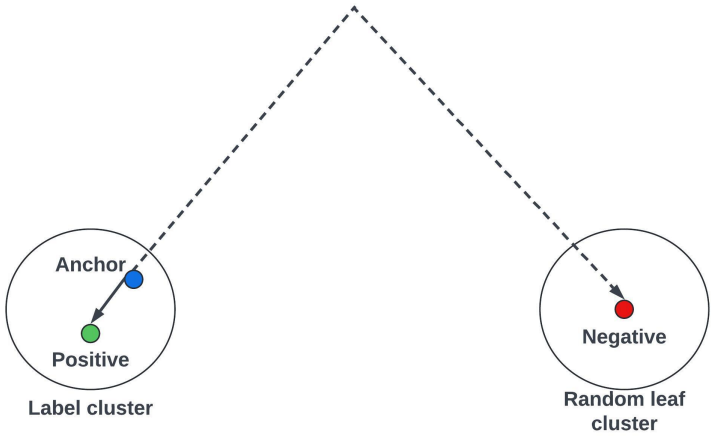

Figure 3.4: Example of Triplet Loss Version 3

3.4.4 Version 4

Assuming that the input is:

Code	Text
J18.9	Male, 80y old: Respiratory failure, unspecified due to Pneumonia, unspecified

Table 3.13: Example of input death certificate in Version 4

The fourth triplet I'm going to implement consists in:

- **Anchor:** Another text with the same label of the input.

Code	Text
J18.9	Female, 47y old: Pneumonia, unspecified

Table 3.14: Example of anchor in Version 4

- **Positive example:** Another text with the same label of the input.

Code	Text
J18.9	Male, 70y old: Sepsis, unspecified due to Pneumonia, unspecified

Table 3.15: Example of positive example in Version 4

- **Negative example:** Another text with a different label from the input (it must be a leaf-label) but not an input label brother.

Code	Text
G11.0	Male, 36y old: Respiratory failure, unspecified due to Congenital nonprogressive ataxia

Table 3.16: Example of negative example in Version 4

This approach has the same functioning as Version 3 (section 3.4.3). The main difference is on the type of approach on the choice of anchor.

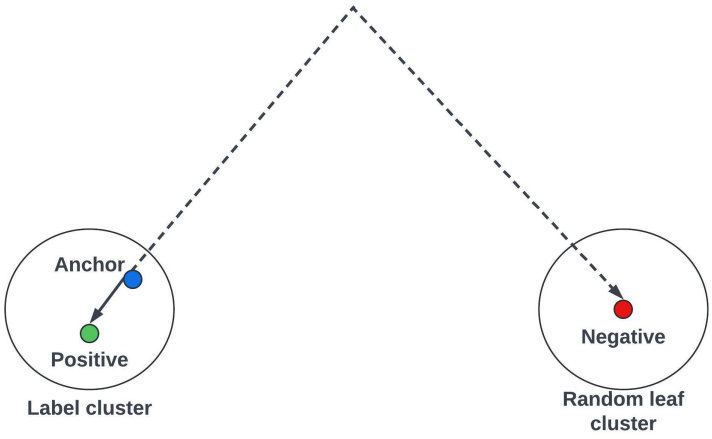

Figure 3.5: Example of Triplet Loss Version 4

3.4.5 Version 5

Assuming that the input is:

Code	Text
J18.9	Male, 80y old: Respiratory failure, unspecified due to Pneumonia, unspecified

Table 3.17: Example of input death certificate in Version 5

The fifth triplet I'm going to implement consists in:

- **Anchor:** The textual description of the input label parent.

Code	Description
J18	Pneumonia, organism unspecified

Table 3.18: Example of anchor in Version 5

- **Positive example:** Another text with the same label of the input.

Code	Text
J18.9	Male, 70y old: Sepsis, unspecified due to Pneumonia, unspecified

Table 3.19: Example of positive example in Version 5

- **Negative example:** Another text that label is a brother of the input.

Code	Text
J18.1	Female, 86y old: Acute respiratory failure due to Lobar pneumonia, unspecified

Table 3.20: Example of negative example in Version 5

Through this approach the label cluster will come closer to the parent cluster and will move away from his brother. Still be the problem seen in section 3.4.2.

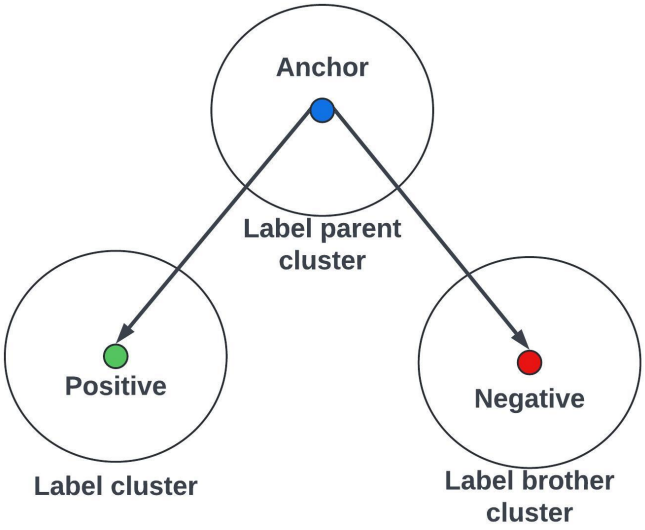

Figure 3.6: Example of Triplet Loss Version 5

3.4.6 Version 6

Assuming that the input is:

Code	Text
J18.9	Male, 80y old: Respiratory failure, unspecified due to Pneumonia, unspecified

Table 3.21: Example of input death certificate in Version 6

This sixth version I'm going to implement consists in no longer a triplet but a in a quartet:

- **Anchor:** The textual description of the input label parent.

Code	Description
J18	Pneumonia, organism unspecified

Table 3.22: Example of anchor in Version 6

- **Positive example:** Another text with the same label of the input.

Code	Text
J18.9	Male, 70y old: Sepsis, unspecified due to Pneumonia, unspecified

Table 3.23: Example of positive example in Version 6

- **Two negative examples:** Other two texts with a different label from the input (it must be a leaf-label) but not an input label brother.

Code	Text
G11.0	Male, 36y old: Respiratory failure, unspecified due to Congenital nonprogressive ataxia
K25.5	Male, 64y old: Gastric ulcer: Chronic or unspecified with perforation

Table 3.24: Example of negative example in Version 6

This version will improve the Version 2 method (section 3.4.2) by distancing more negative examples from the anchor label.

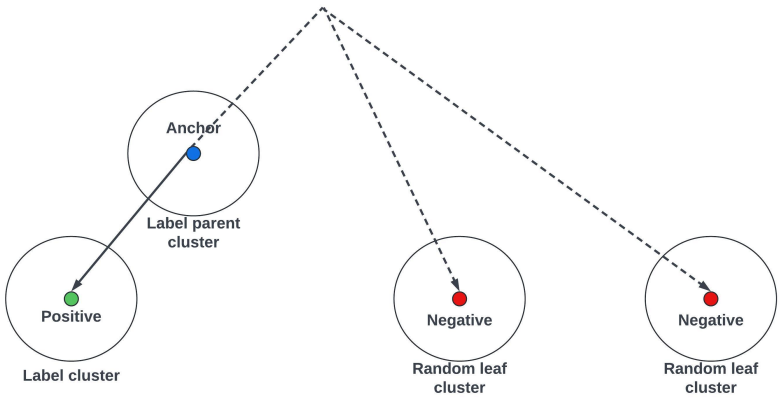

Figure 3.7: Example of Triplet Loss Version 6

3.5 Dataset Variation 1

The first dataset variation will consist of re-train and re-test the three best performing algorithms with a modification to the datasets. In the two datasets will be removed all the ", unspecified" strings from the texts. This is because such string can lead to some misunderstandings from the model in the unspecified labels classification.

Text
Female, 86y old: Acute respiratory failure due to Lobar pneumonia, unspecified

Table 3.25: Example of Base Dataset

↓

Text
Female, 86y old: Acute respiratory failure due to Lobar pneumonia

Table 3.26: Example of Dataset Variation 1

3.6 Dataset Variation 2

The second dataset variation will consist of re-train and re-test the three best performing algorithms with a modification to the datasets. In the two datasets will be removed all the unspecified labels, all the leafs that end in ".9" The motivation is the same as Dataset Variation 1 (section 3.5) but with a different approach.

Code	**Text**
X44	Male, 30y old: (Poisoning: Other and unspecifi...
M35.9	Female, 39y old: Other secondary pulmonary hyp...
K25.5	Female, 68y old: Sepsis, unspecified due to Ga...
C67.9	Male, 63y old: Cardiac arrest, unspecified due...
J18.9	Male, 80y old: Respiratory failure, unspecifie...
...	

Table 3.27: Example of Base Dataset

↓

Code	**Text**
X44	Male, 30y old: (Poisoning: Other and unspecifi...
K25.5	Female, 68y old: Sepsis, unspecified due to Ga...
W19	Male, 96y old: Fracture of femur, part unspeci...
F03	Female, 85y old: Congestive heart failure due ...
I25.0	Female, 87y old: Stroke, not specified as haem...
...	

Table 3.28: Example of Dataset Variation 2

3.7 Final Dataset Variation

The final dataset variation will consist of re-train and re-test the best performing algorithm and the base model with larger datasets, respectively of 400000 and 100000 certificates. The purpose of this variation will be to compare the results in a bigger scale in order to observe the final improvements.

4

Results

In this fourth chapter I'm going to present the results of the six versions and three variations previously presented in chapter 3. Firstly, in section 4.1 at the leaf-level, then in section 4.2 at the category level. The calculated metrics will be: Accuracy (Acc), Macro/Micro/Weighted F1 (McF1,MiF1,WF1), Macro/Micro/Weighted Precision (McP,MiP,WP) and Macro/Micro/Weighted Recall (McR,MiR,WR). For the purpose of this experiment, the most important metric I'm going to consider is accuracy. Secondly, I will present the main errors of the versions and the first two dataset variations (section 4.3). Finally, in section 4.4 I'm going to show all the results of the training of the best performing algorithm, compared to the base model, on larger datasets. In addition, in section 4.5 I'm going to present some statistics about the used architecture and the computation times.

4.1 Comparisons

4.1.1 Version Comparison

The training and testing of the six versions shows that the Version 4 (section 3.4.4) is the best performing algorithm for the leaf level classification, reaching an accuracy of 0.876. The best improvement of this method is that outperforms the Cross Entropy Loss (section 3.3) by 0.015 (4.1% more), 0.025 (7.3% more) and 0.024 (5.7% more) respectively in Macro F1, Macro Precision and Macro Recall.

The other two best performing algorithms are the Second (section 3.4.2) and Sixth (section 3.4.6) version, that have both reached an accuracy of 0.874, 0.002 less than the best. They also reached good improvements in Macro F1, Macro Precision and Macro Recall but not as great as the best one.

Metric	AC	McF1	McP	McR	MiF1	MiP	MiR	WF1	WP	WR
B	0.870	0.363	0.342	0.416	0.870	0.870	0.870	0.838	0.819	0.870
Ds1	0.872	0.379	0.355	0.437	0.872	0.872	0.872	0.842	0.824	0.872
Ds2	0.874	0.382	0.359	0.436	0.874	0.874	0.874	0.843	0.824	0.874
Ds3	0.872	0.370	0.347	0.424	0.872	0.872	0.872	0.840	0.820	0.872
Ds4	**0.876**	**0.388**	**0.367**	**0.440**	**0.876**	**0.876**	**0.876**	**0.845**	**0.827**	**0.876**
Ds5	0.872	0.359	0.337	0.414	0.872	0.872	0.872	0.841	0.822	0.872
Ds6	0.874	0.378	0.356	0.431	0.874	0.874	0.874	0.843	0.823	0.874

Table 4.1: Version Comparison

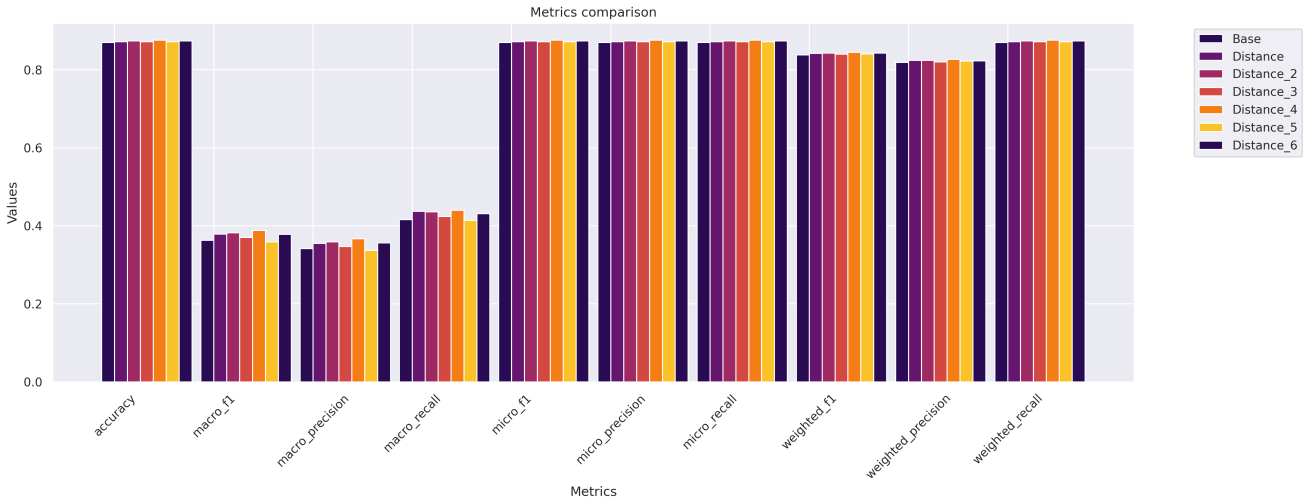

Figure 4.1: Metrics Comparison

The Fourth Version have the highest values in all the metrics but it has reached only a Macro-F1 of 0.388, a Macro-Precision of 0.367 and a Macro-Recall of 0.440. Low values in the Macro metrics is a common problem in all confrontations and is due to the highest number of death from certain diseases. This leads to an imbalance in the number of instances in the train and test datasets between some classes.

4.1.2 Dataset Variation 1 Comparison

Starting from the three best performing algorithms, Dataset Variation 1 (section 3.5) has shown that there is an improvement in the classification by removing the ", unspecified" string from the text. This proves that those strings create some misunderstandings in the model but are not necessary for classification purpose.

Mt	AC	McF1	McP	McR	MiF1	MiP	MiR	WF1	WP	WR
DV1:Ds2	**0.880**	0.395	0.371	0.450	**0.880**	**0.880**	**0.880**	0.849	0.828	**0.880**
DV1:Ds4	0.874	0.382	0.359	0.435	0.874	0.874	0.874	0.842	0.821	0.874
DV1:Ds6	**0.880**	**0.403**	**0.382**	**0.456**	**0.880**	**0.880**	**0.880**	**0.850**	**0.832**	**0.880**

Table 4.2: Dataset Variation 1 Comparison

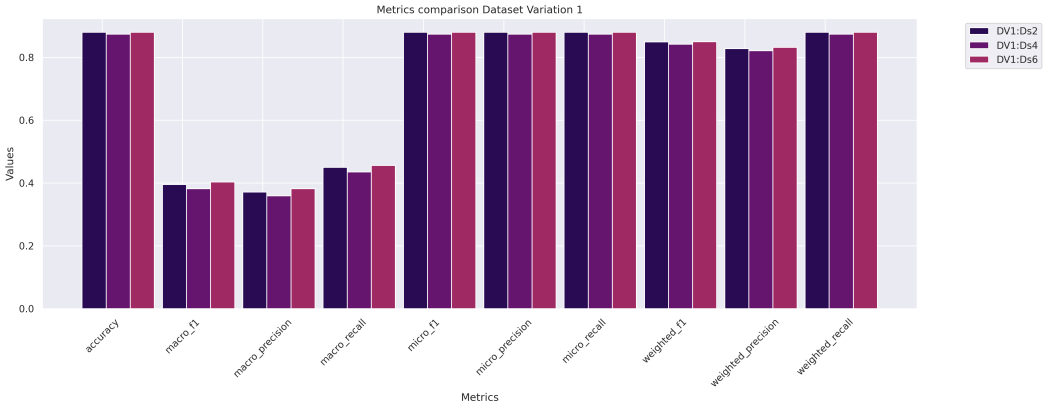

Figure 4.2: Metrics Comparison Dataset Variation 1

After this Variation, in Version 4 (section 3.4.4) there's a worsening in accuracy of 0.002. On the other hand, in Version 2 and 4 (section 3.4.2 and 3.4.4) there is an improvement by 0.006, from 0.874 to 0.880, with some marginal improvements in the Sixth version over the Second in some other metrics.

4.1.3 Dataset Variation 2 Comparison

The Second Variation (section 3.6) has shown that in classification of labels other than unspecified the best performing algorithms are the Second (section 3.4.2) and Fourth (section 3.4.4) Version. They have reached an accuracy of 0.835, with some marginal improvements in the Second version over the Fourth in some other metrics.

Mt	AC	McF1	McP	McR	MiF1	MiP	MiR	WF1	WP	WR
DV2:Ds2	**0.835**	**0.328**	**0.306**	**0.388**	**0.835**	**0.835**	**0.835**	**0.795**	**0.771**	**0.835**
DV2:Ds4	**0.835**	0.317	0.294	0.377	**0.835**	**0.835**	**0.835**	0.793	0.769	**0.835**
DV2:Ds6	0.824	0.296	0.274	0.353	0.824	0.824	0.824	0.782	0.758	0.824

Table 4.3: Dataset Variation 2 Comparison

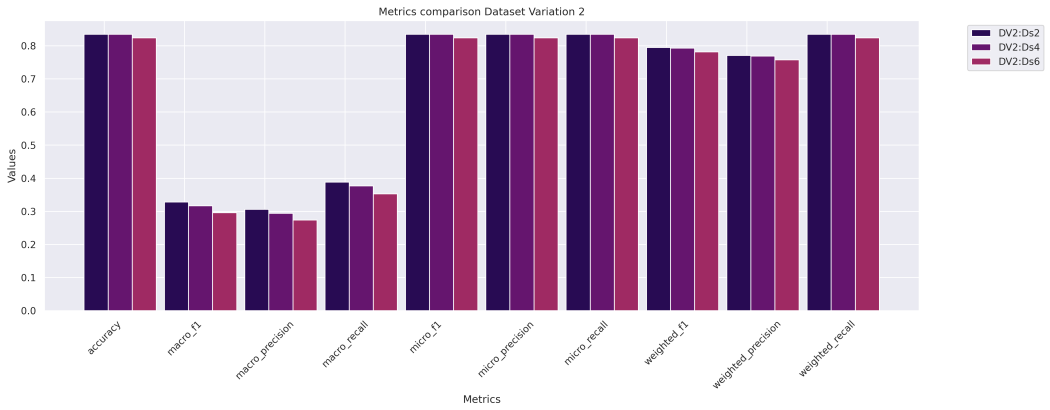

Figure 4.3: Metrics Comparison Dataset Variation 2

This variation is not comparable with the previous variation or the other Triplet Versions due to his reduction in the datasets. Nevertheless, this variation shows that for the automatic classification without the unspecified labels the best algorithm still the same as the Dataset Variation 1 (section 3.5).

4.2 Comparisons on category level

4.2.1 Version Comparison on category level

Looking at the metrics on the category level classification the best performing algorithm is the Fourth (section 3.4.4), similarly to the leaf-level classification, reaching an accuracy of 0.897. Moreover, the Second (section 3.4.2) outperforms the Fourth in some metrics.
The three best performing algorithms remain the Second (section 3.4.2) , the Fourth (section 3.4.4) and Sixth (section 3.4.6) Version, that have reached an accuracy of 0.897.

Mt	AC	McF1	McP	McR	MiF1	MiP	MiR	WF1	WP	WR
B	0.892	0.458	0.451	0.494	0.892	0.892	0.892	0.871	0.860	0.892
Ds1	0.892	0.476	0.468	0.515	0.892	0.892	0.892	0.873	0.863	0.892
Ds2	0.896	**0.491**	**0.486**	**0.525**	0.896	0.896	0.896	**0.877**	**0.867**	0.896
Ds3	0.896	0.477	0.471	0.510	0.896	0.896	0.896	0.875	0.864	0.896
Ds4	**0.897**	0.486	0.481	0.520	**0.897**	**0.897**	**0.897**	**0.877**	0.866	**0.897**
Ds5	0.893	0.455	0.451	0.492	0.893	0.893	0.893	0.872	0.862	0.893
Ds6	0.896	0.463	0.452	0.501	0.896	0.896	0.896	0.876	0.864	0.896

Table 4.4: Version Comparison on Category Level

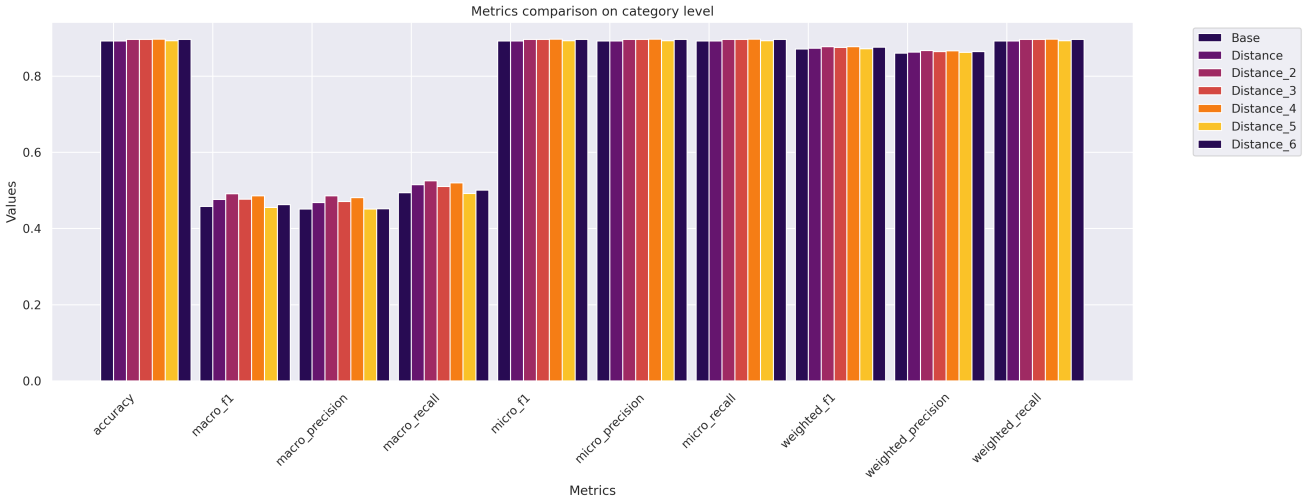

Figure 4.4: Metrics Comparison on Category Level

This comparison shows that the classification on the category level has a higher accuracy due to some misclassification between brother labels. We can observe this improvement more in some metrics than the others. As instance, in Macro F1, Macro Precision and Macro Recall there is an improvement respectively of 0.031 (6.8% more), 0.035 (7.8% more) and 0,031 (6.3% more).

4.2.2 Dataset Variation 1 Comparison on category level

Starting from the three best performing algorithms, the results of Dataset Variation 1 (section 4.1.2) at the leaf-level has shown that there is an improvement in the classification by removing the ", unspecified" string from the text. This improvement is also transmitted at the category level classification. In fact, the Second Version (section 3.4.2) has reached an accuracy of 0.902, the best observed so far.

Mt	AC	McF1	McP	McR	MiF1	MiP	MiR	WF1	WP	WR
DV1:Ds2	**0.902**	**0.501**	**0.499**	0.535	**0.902**	**0.902**	**0.902**	**0.884**	**0.875**	**0.902**
DV1:Ds4	0.897	0.478	0.473	0.514	0.897	0.897	0.897	0.876	0.865	0.897
DV1:Ds6	0.901	0.497	0.490	**0.537**	0.901	0.901	0.901	0.881	0.871	0.901

Table 4.5: Dataset Variation 1 Comparison on Category Level

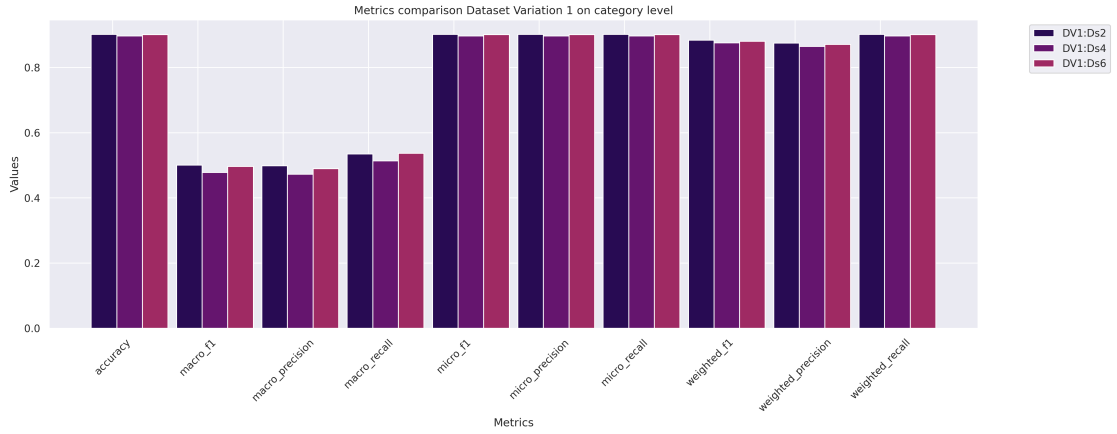

Figure 4.5: Metrics Comparison Dataset Variation 1 on Category Level

4.2.3 Dataset Variation 2 Comparison on category level

The Second Variation (section 3.6) at the category level classification also has improvements in terms of accuracy and other metrics, reaching an accuracy of 0.857 with the Fourth Version (section 3.4.2).

Mt	AC	McF1	McP	McR	MiF1	MiP	MiR	WF1	WP	WR
DV2:Ds2	0.856	**0.415**	**0.402**	**0.463**	0.856	0.856	0.856	**0.826**	**0.811**	0.856
DV2:Ds4	**0.857**	0.400	0.387	0.448	**0.857**	**0.857**	**0.857**	0.825	0.810	**0.857**
DV2:Ds6	0.846	0.380	0.367	0.430	0.846	0.846	0.846	0.814	0.798	0.846

Table 4.6: Dataset Variation 2 Comparison on Category Level

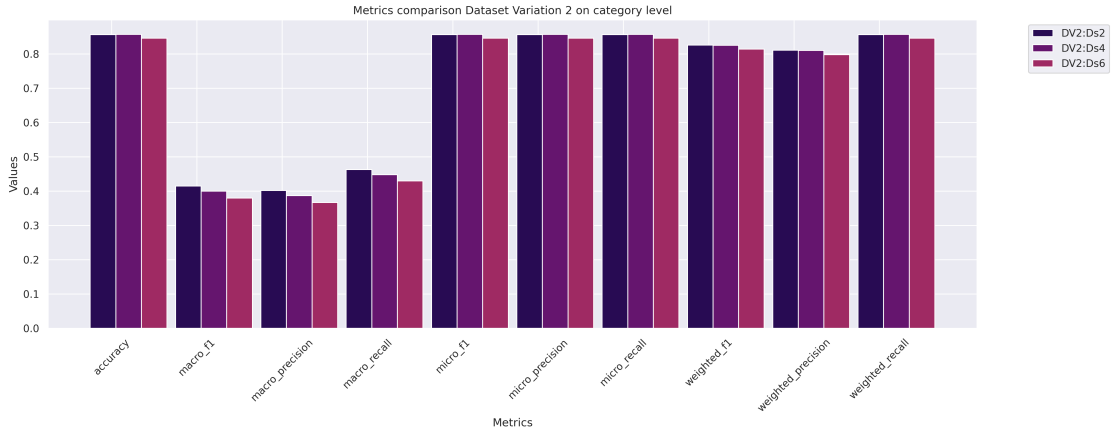

Table 4.7: Metrics Comparison Dataset Variation 2 on Category Level

4.3 Main errors

4.3.1 Main version errors

The main errors are in classes J44.9, I25.1, C97 and J18.9 . In every Version the majority of errors are in unspecified classes. This led me to the creation of the Dataset Variation 1 and 2 (section 3.5 and 3.6) to try to solve this problem.

Code	B	Code	Ds1	Code	Ds2	Code	Ds3	Code	Ds4	Code	Ds5	Code	Ds6
J44.9	10	J44.9	8	J18.9	8	J44.9	9	J44.9	10	J44.9	9	J44.9	5
I25.1	7	I25.1	8	J44.9	7	C97	7	C97	7	J18.9	6	J18.9	5
J18.9	6	C97	7	C97	6	J18.9	6	I25.1	5	I25.1	6	C97	4
C97	6	J18.9	6	I25.1	5	I25.1	6	J18.9	4	C97	5	I26.9	4
K70.4	3	G30.9	6	I21.9	5	E43	5	E14.7	3	I26.9	4	I10	3

Table 4.8: Main Version Errors

4.3.2 Main Dataset Variation 1 errors

After the application of the Dataset Variation 1 still some problems with the class J44.9, J18.9, I25.1 and C97 but only the first two classes are unspecified labels. Looking at how these two classes are coded, I discovered that they are not coded like other unspecified classes with the ", unspecified" string at the end:

- Female, 76y old: Respiratory failure ... *Unspecified mental and behavioural disorder*

This different type of coding the text is the reason why still the misclassification error in those classes.

Code	DV1:Ds2	Code	DV1:Ds4	Code	DV1:Ds6
J44.9	9	J44.9	10	J44.9	9
I25.1	6	C97	7	C97	6
C97	5	J18.9	6	I26.9	4
I10	4	I25.1	6	J18.9	4
E14.7	4	I10	5	I25.1	4

Table 4.9: Main Dataset Variation 1 Errors

4.3.3 Main Dataset Variation 2 errors

After the application of Dataset Variation 2 the main problems still with the class C97, I25.1 and others. Analysing this two classes, I've noticed that the class C97, *Malignant neoplasms of independent (primary) multiple sites*, is a generic class comparable to an unspecified class. Subsequently, examining the class I25.1, *Atherosclerotic heart disease*, I've discovered that is an unspecified class without the ending ".9":

- **C97:** Female, 72y old: Respiratory failure, *unspecified due to Pulmonary oedema due to ...*
- **I25.1:** Male, 75y old: Cardiac arrest, *unspecified due to Chronic kidney disease ...*

Code	DV2:Ds2	Code	DV2:Ds4	Code	DV:Ds6
C97	4	I25.1	5	I25.1	8
I25.1	4	E14.7	4	C97	5
E14.7	4	C97	4	E14.7	4
E11.7	3	I69.4	3	I69.4	3
J10.0	3	J10.0	3	J10.0	3

Table 4.10: Main Dataset Variation 2 Errors

4.4 Final Dataset Variation Results

As a result of the experiments in section 4.1 and 4.2, the top-performing algorithm is the one observed with Dataset Variation 1 (section 3.5). After training and testing on larger datasets this algorithm and the base model, the metrics achieved at the leaf-level classification are:

Mt	AC	McF1	McP	McR	MiF1	MiP	MiR	WF1	WP	WR
FDV:B	0.964	0.438	0.432	0.460	0.964	0.964	0.964	0.958	0.954	0.964
FDV:Ds	**0.966**	**0.443**	**0.437**	**0.465**	**0.966**	**0.966**	**0.966**	**0.960**	**0.956**	**0.966**

Table 4.11: Final Dataset Variation Comparison

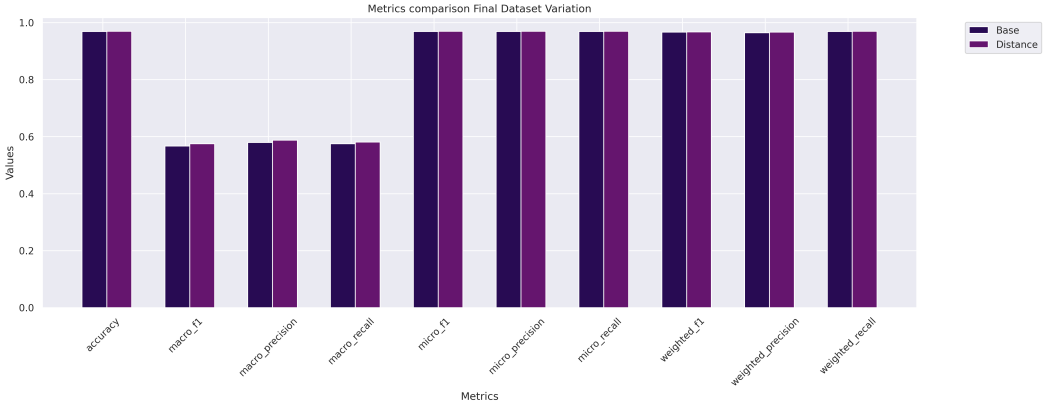

Figure 4.6: Metrics Comparison Final Dataset Variation 1

Then, with regard to classification at category level, the metrics obtained are:

Mt	AC	McF1	McP	McR	MiF1	MiP	MiR	WF1	WP	WR
FDV:B	0.969	0.567	0.580	0.575	0.969	0.969	0.969	0.967	0.965	0.969
FDV:Ds	**0.970**	**0.575**	**0.588**	**0.581**	**0.970**	**0.970**	**0.970**	**0.968**	**0.967**	**0.970**

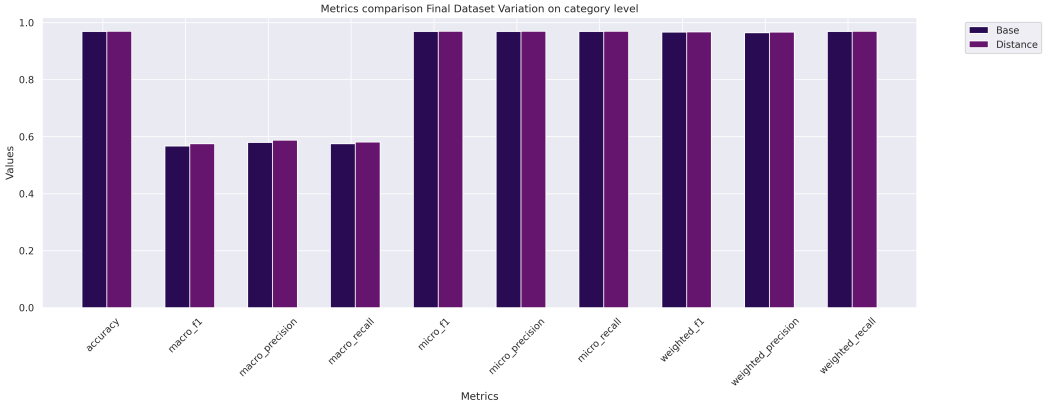

Figure 4.7: Metrics Comparison Final Dataset Variation on Category Level

The main errors observed are in classes I50.0, I25.1, J44.9, J18.9 .

Code	FDV:B	Code	FDV:Ds
I50.0	75	I50.0	68
I25.1	60	J44.9	59
J44.9	59	I25.1	55
J18.9	45	J18.9	42
I64	41	I48.9	38

Table 4.12: Final Dataset Variation Errors

The distribution of errors among the classes shows that only some classes have a large number of errors, while the others have between 1 and 10.

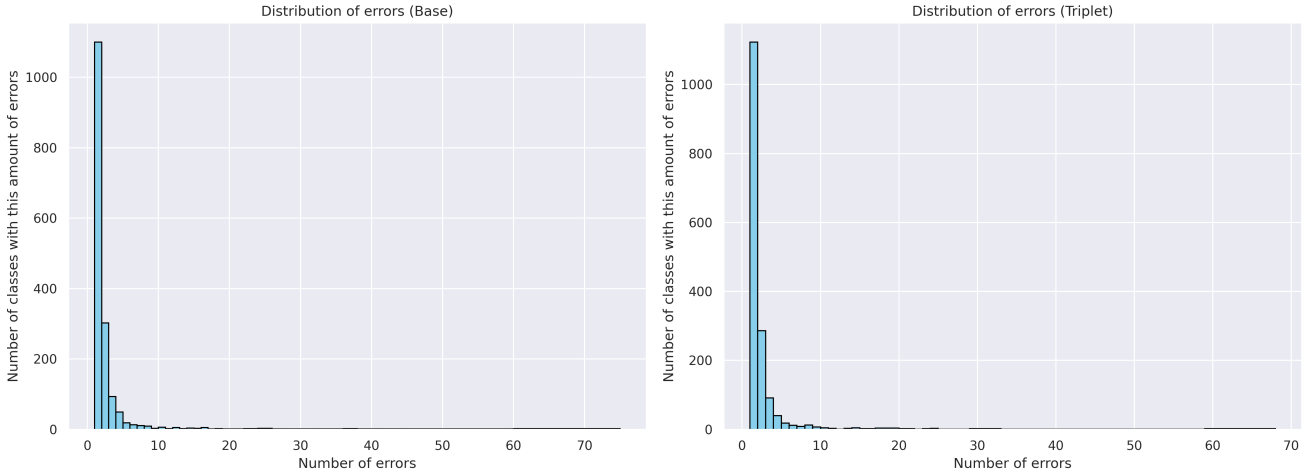

Figure 4.8: Final Dataset Variation Distribution

In conclusion, there are some improvements with the new algorithm that outperforms the base metric. This improvements are not very evident as in previous comparisons due to the high size of the datasets. However, the information obtained from this final dataset variation confirm what was observed in the past sections.

4.5 Computation Time

To give some statistics, the BERT-Base-Uncased model takes around 2 hours to train and 1 minute to test ,with the smaller datasets, on a machine with a Nvidia Titan XP GPU and 80GB of RAM memory. With the larger datasets, on the same architecture, the training takes around 20 hours and the testing 15 minutes.

5

Discussion and Conclusion

5.1 Discussions

In a view of future research on improving this method, the focus could be on improving the selection of the triplet used in the calculation of the loss value (section 3.4) through advanced sampling methods. Furthermore, in order to help the model better understand death certificates for specific classes, two options are possible: (1) improve the method observed in Dataset Variation 1 (section 3.5) by trying to encode the classes better or (2) build two distinct classifiers for classical and unspecified labels.

5.2 Conclusions

ICD's are a very helpful standard for coding death certificates. Nevertheless, there's a big amount of manual work in the hands of manual coders. In order to reduce this problem, the automatic classification for death certificates needs an improvement. The proposed research has shown that using a BERT-Base-Uncased model with an innovative ontology-based approach seems to outperform the effectiveness of the state-of-art models. Using a text coded without the string ", unspecified" and a triplet formed by the input label parent as anchor, an input label twin as positive example and another label different from input label brothers as negative example gives the best results, reaching an Accuracy of 96.6% on the leaf level and of 97.0% at the category level. The most noticeable improvement is on the macro metrics at the category level, reaching a Macro F1 of 0.575, a Macro Precision of 0.588 and a Macro Recall of 0.581. Generally, in the macro-metrics there is an improvement slightly more than 1%, while an improvement of 0.1% is recorded in the other metrics.

Bibliography

[1] Enrique Amigo and Agustín Delgado. Evaluating extreme hierarchical multi-label classification. *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5809–5819, May 2022.

[2] AssemblyAI. The Full Story of Large Language Models and RLHF, accessed Apr 2024.

[3] Y. Baghdadi, A. Bourree, A. Robert, G. Rey, A. Gallay, P. Zweigenbaum, C. Grouin, and A. Fouillet. Automatic classification of free-text medical causes from death certificates for reactive mortality surveillance in france. *International Journal of Medical Informatics*, 131:103915, 2019.

[4] S Belamri, Achille Aouba, G Pavillon, and E Jougla. Connaissance des causes de décès en algérie. Étude des décès enregistrés par l'insp. méthodes et premiers résultats. *Revue d'épidémiologie et de santé publique*, 58:226–30, 06 2010.

[5] Christopher Groeneveld. Simple ICD-10, accessed Mar 2024.

[6] Eduardo P. Costa, Ana C. Lorena, Andre C.P.L.F. Carvalho, and Alex A. Freitas. A review of performance evaluation measures for hierarchical classifiers. *AAAI Workshop - Technical Report*, 2007.

[7] Vincenzo Della Mea, Mihai Horia Popescu, and Kevin Roitero. Underlying cause of death identification from death certificates using reverse coding to text and a nlp based deep learning approach. *Informatics in Medicine Unlocked*, 21:100456, 2020.

[8] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *North American Chapter of the Association for Computational Linguistics*, 2019.

[9] F. Duarte, B. Martins, C.S. Pinto, and M.J. Silva. Deep neural models for icd-10 coding of death certificates and autopsy reports in free-text. *Journal of Biomedical Informatics*, 80:64–77, 2018.

[10] O Eckert. Elektronische Kodierung von Todesbescheinigungen [Electronic coding of death certificates]. *Bundesgesundheitsblatt Gesundheitsforschung Gesundheitsschutz*, 62(12):1468–1475, Dec 2019.

[11] Laurent Falissard, Célia Morgand, Stéphanie Roussel, Camille Imbaud, Walid Ghosn, Khaled Bounebache, and Grégoire Rey. A deep artificial neural network-based model for prediction of underlying cause of death from death certificates: Algorithm development and validation. *JMIR Medical Informatics*, 8(4):e17125, 2020.

[12] P Harteloh. The implementation of an automated coding system for cause-of-death statistics. *Inform Health Soc Care*, 45(1):1–14, Jan 2020.

[13] International Classification of Diseases (ICD). ICD-10 Browser, accessed Apr 2024.

[14] RA Israel. Automation of mortality data coding and processing in the United States of America. *World Health Stat Q*, 43(4):259–262, 1990.

[15] B. Koopman, G. Zuccon, A. Nguyen, A. Bergheim, and N. Grayson. Automatic icd-10 classification of cancers from free-text death certificates. *International Journal of Medical Informatics*, 84(11):956–965, 2015.

[16] Aris Kosmopoulos, Ioannis Partalas, Eric Gaussier, Georgios Paliouras, and Ion Androutsopoulos. Evaluation measures for hierarchical classification: a unified view and novel approaches. *Data Mining and Knowledge Discovery*, 29, 2013.

[17] Anqi Mao, Mehryar Mohri, and Yutao Zhong. Cross-entropy loss functions: theoretical analysis and applications. *Proceedings of the 40th International Conference on Machine Learning*, 2023.

[18] Zulfat Miftahutdinov and E. Tutubalina. Kfu at clef ehealth 2017 task 1: Icd-10 coding of english death certificates with recurrent neural networks. *Conference and Labs of the Evaluation Forum*, 2017.

[19] Oxford Semantic Technologies. Oxford semantic technologies, 2024.

[20] Mihai Horia Popescu, Kevin Roitero, Stefano Travasci, and Vincenzo Della Mea. Automatic assignment of icd-10 codes to diagnostic texts using transformers based techniques. *2021 IEEE 9th International Conference on Healthcare Informatics (ICHI)*, pages 188–192, 2021.

[21] PyTorch Contributors. TripletMarginWithDistanceLoss, accessed Mar 2024.

[22] Rete delle Classificazioni. Classificazioni ICD-9-CM e ICD-10, accessed Apr 2024.

[23] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[24] Rudi Studer, V.Richard Benjamins, and Dieter Fensel. Knowledge engineering: Principles and methods. *Data Knowledge Engineering*, 25(1):161–197, 1998.

[25] Fan Teng, Zhiqiang Ma, Jing Chen, Ming Xiao, and Liang Huang. Automatic Medical Code Assignment via Deep Learning Approach for Intelligent Healthcare. *IEEE J Biomed Health Inform*, 24(9):2506–2515, Sep 2020.

[26] University of Udine. Data Techniques for E-Health, accessed Apr 2024.

[27] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems*, 2023.

[28] Matthew West. 6 - some general principles for conceptual, integration, and enterprise data models. In Matthew West, editor, *Developing High Quality Data Models*, pages 63–78. Morgan Kaufmann, Boston, 2011.