NATIONAL AND KAPODISTRIAN UNIVERSITY OF ATHENS

SCHOOL OF SCIENCE
DEPARTMENT OF INFORMATICS AND TELECOMMUNICATION

MSc THESIS

Multi-Label Text Classification for Greek Legal Documents

Apostolos N. Papatheodorou

Supervisor: Manolis Koubarakis, Professor

Athens
December 2021

ΕΘΝΙΚΟ ΚΑΙ ΚΑΠΟΔΙΣΤΡΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Ταξινόμηση Κειμένων Πολλαπλών Ετικετών Της Ελληνικής Νομοθεσίας

Απόστολος Ν. Παπαθεοδώρου

ΕΠΙΒΛΕΠΩΝ: Μανόλης Κουμπαράκης, Καθηγητής

Αθήνα
Δεκέμβριος 2021

MSc THESIS

Multi-Label Text Classification for Greek Legal Documents

Apostolos N. Papatheodorou

ID: DS1190015

SUPERVISOR: **Manolis Koubarakis,** Professor

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Ταξινόμηση Κειμένων Πολλαπλών Ετικετών Της Ελληνικής Νομοθεσίας

Απόστολος Ν. Παπαθεοδώρου

ΑΜ: DS1190015

ΕΠΙΒΛΕΠΩΝ: **Μανόλης Κουμπαράκης,** Καθηγητής

ABSTRACT

Extreme Multi-label Text Classification (XMTC) refers to the problem of tagging text sequences with a set of labels extracted from a large collection of hundreds or even thousands of concepts. The scientific field of XMTC has received substantial attention in the last few years since it can be found in many industrial and real-world scenarios. Biomedical text annotation, Wikipedia page tagging, amazon product labeling, and legal document categorization are only some of its most widespread commercial applications. In this dissertation, we are addressing the multi-label text classification problem based on legal documents derived from the legislation code of the Greek parliament.

To achieve our goals, we use the *Raptarchis47k dataset*, a novel hierarchically structured dataset that encompasses Greek legal texts from the period between 1835 to 2015. Unfortunately, because *Raptarcihis47k* is recently released little is known about its inner structure and composition. Despite this, however, during the writing of our thesis, we discover some useful but unknown information about its label connections and hierarchy. Hence, we provide a thorough and concise analysis of the intrinsic structure and organisation of the dataset.

Going forward and after the end of the above procedure we continue with the development of our models. So, considering that deep learning approaches and especially transformers have achieved extraordinary results on many downstream NLP tasks, we try to take advantage of these methods and generalize them to the requirements our problem. Nevertheless, having noticed that naively applied BERT-based models directly to XMTC leads to sub-optimal solutions, we follow as guideline other state-of-the-art methods in the field. Finally, We also give our suggestions and proposals in order to deal with the problem in the best possible and more efficient way.

All things considered, we anticipate this thesis to contribute to the expansion of the existing Greek NLP literature and also, to establish a solid groundwork for any future study relevant to the domain of legal multi-label text classification.

SUBJECT AREA: Deep Learning, Natural Language Processing

KEYWORDS: BERT Based Models, Extreme Multi-label Classification

ΠΕΡΙΛΗΨΗ

Η *Ταξινόμηση Κειμένου Πολλαπλών Ετικετών της Ελληνικής Νομοθεσίας* αναφέρεται στο πρόβλημα της αντιστοίχισης ενός κειμένου με τις πιο σχετικές κατηγορίες απο ενα συνολο εκατοντάδων ή και χιλιάδων ετικετών. Ο ερευνιτικός αυτός κλάδος έχει λάβει αρκετή προσοχή τα τελευταία χρόνια αφου συναντάται σε πολλές εφαρμογές και προβλήματα τόσο στον πραγματικό κόσμο όσο και στον βιομηχανικό τομέα. Η κατηγοριοποίηση βιο-ιατρικών κειμένων, ιστοσελίδων της Wikipedia, προϊόντων της Amazon, και κειμένων νομικού περιεχομένου, είναι μονο μερικά χαρακτηριστικά παραδείγματα. Σε αυτήν την διπλωματική εργασία μελετάμε το παραπάνω πρόβλημα υπό το πρίσμα κειμένων νομοθετικού περιεχομένου τα οποία προέρχονται κατευθείαν απο το ελληνικό κοινοβούλιο.

Για τα πειράματα μας χρησιμοποιήσαμε το *Raptarchis47k*, ένα νέο ιεραρχικά δομημένο σύνολο δεδομένων (dataset) που περιέχει αρχεία τού ελληνικού συντάγματος της περιόδου 1832-2015. Επιπλέον επειδή το συγκεκριμένο dataset έχει δημοσιευθεί αρκετά πρόσφατα, λίγα είναι γνωστά για την φύση και σύνθεση του. Παρόλα αυτά κατά την συγγραφή αυτης της εργασίας, ανακαλύψαμε νέες πληροφορίες σχετικές με την ιεραρχια και τη οργάνωση του. Ως αποτέλεσμα λοιπον, παραθέτουμε μια περιεκτική ανάλυση της δομής και των αλληλοσυνδέσεων των labels (ετικετών) που υπάρχουν σε αυτό.

Έπειτα συνεχίζουμε με την ανάπτυξη των μοντέλων μας. Συλλογιζόμενοι τα εξαιρετικά αποτελέσματα της Βαθιάς Μηχανικής Μάθησης, επιχειρούμε να επωφεληθούμε απο αυτές τις προσεγγίσεις προσπαθώντας να τις ενσωματώσουμε στις απαιτήσεις του προβλήματος. Εντούτοις, ξέροντας πως η αφελής, βιαστική και απλοική εφαρμογή τους οδηγει σε υπο-βέλτιστα αποτελέσματα, ακολουθούμε σαν πυξιδα τις σημαντικότερες δημοσιεύσεις στην περιοχή, ενω παράλληλα συζητάμε και προτείνουμε τις δικες μας λύσεις και προτασεις.

Συνολικά, προσδοκούμε αυτή η μελέτη να συνεισφέρει στην υπάρχουσα βιβλιογραφία στον τομέα της *Επεξεργασίας Φυσικής Γλώσσας (NLP)* καθώς επίσης και να αποτελέσει εφαλτήριο για περαιτέρω μελέτη, κατανόηση και επίλυση του τρέχοντος προβλήματος.

SUBJECT AREA: Βαθιά Μηχανική Μάθηση, Επεξεργασία Φυσικής Γλώσσας

KEYWORDS: BERT Μοντέλα, Ταξινόμηση Κειμένων Πολλαπλών Κατηγοριών

To my family

”The limits of my language means the limits of my world”

— Ludwig Wittgenstein

ACKNOWLEDGEMENTS

The writing of this master thesis was a challenging and time-consuming task. However, thanks to the contribution of several people it was possible to successfully complete this significant for me piece of work. Thus, at this point, I would like to thank my supervisor Professor Manolis Koubarakis for allowing me to work on a so interesting and exciting NLP problem. This experience, along with his courses, was a source of inspiration for me as it gave me the opportunity to explore and better understand many leading-edge areas of Artificial Intelligence and Deep Learning. Secondly, I wish to thank Despina Pantazi and Christos Papaloukas for their constant assistance, guidance, and patience during the writing of this dissertation. Finally, I want to express my gratitude to my parents and my sister for their unconditional encouragement and support during all the years of my studies.

CONTENTS

LIST OF FIGURES

70figure.caption.41
71figure.caption.43

LIST OF TABLES

LIST OF ALGORITHMS

PREFACE

The current diploma thesis is submitted for the acquisition of a MSc degree from the postgraduate program of *Data Science and Information Technologies (DSIT)* which is organized and administrated by the *Department of Informatics and Telecommunications* of *NKUA*. All the work presented henceforth is related to the area of *Big Data and Artificial Intelligence* (specialization-1 of the DSIT program) and particularly with the sub-field of Natural Language Processing (NLP). Additional information complementary to the context can be found in the appendix at the end of this dissertation.

1. INTRODUCTION

1.1 General Introduction

In the past several years a lot of progress has been made in the fields of Machine Learning (ML) and Artificial Intelligence (AI). Particularly, in the area of Natural Language Processing (NLP) numerous influential approaches [3, 7–10] have brought new state-of-the-art results in many downstream tasks like Name Entity Recognition (NER), Question Answering (QA), Sentiment Analysis (SA), and Text Categorisation (TC). The great improvements and results in such complex problems have led to the extensive use of Deep Learning (DL) solutions not only by the scientific community, but also by the industrial and business world. Nowadays, Artificial Intelligence is considered a significant tool competent enough to analyze vast amounts of data, spare time, and release valuable human resources. Organizations from different fields like medicine, retail outlets, or legal services strive to take advantage of the contemporary AI systems in order to avoid potential risks and improve the quality of their products.

One of the areas which have received a lot of attention from both professionals and AI researchers is the field of legal NLP [11–14]. This happens mainly due to the fact that the majority of the resources in legal domain are provided at textual data such as contracts, law records, and official judgments. As a result, several novel datasets [15–18] have been proposed to promote and facilitate the development of algorithms for legal purposes. Most of these methods usually confront complex problems like text summarization, judgment predictions, text categorization, etc. In Fig-1.1 we display a more detailed division of the legal NLP tasks according to [12].

In this dissertation, we will focus on the problem of legal Extreme Multi-Lable Text Classification (XMTC) problem. An extremely challenging topic with many real-life daily activities and applications in the legal domain.

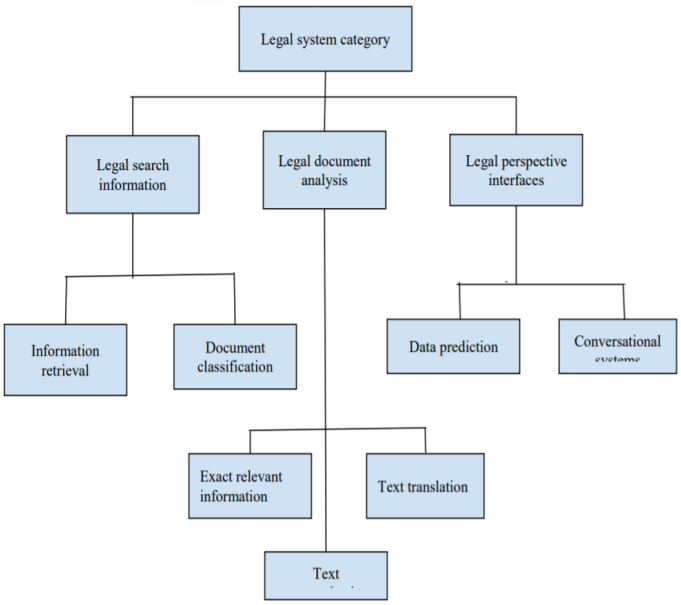


Figure 1.1: Legal Domain categories

1.2 Extreme Multi-Label Text Classification

In a general sense, XMTC refers to the task of assigning a given document with its most relevant labels from a large-scale set of hundreds or even thousands of concepts. XMTC has become increasingly important since it can be found in various forms, such as item categorization, web page tagging, and biomedical text annotation. Recently, the rise of deep pre-trained transformer models like BERT in combination with the advent of numerous innovative approaches (see Chapter 2) has led to new breakthroughs and benchmarks in the topic. Unfortunately, despite the constant progress and the great number of publications, these techniques have been primarily applied to resource-rich languages like English. Meanwhile, the XMTC literature for less prevalent languages like Greek still remains rather limited and obscure.

In this thesis, in an attempt to solve the above issues and enhance the existing Greek NLP bibliography, we engage with the task of extreme multi-label classification based on a corpus of Greek legal documents. More specifically, our target is to solve the XMTC problem by utilizing BERT-based models and incorporate them into a unified framework with different techniques (along with our suggestions and proposals) that are capable to exploit the label hierarchy of the dataset and take into account the intrinsic architecture of BERT. The findings indicate that in this way, we can obtain satisfactory results, while at the same time we achieve better fine-tuning and parameter utilization for our models.

For the experiments, we used the Raptarchis47k dataset, a novel dataset that holds the main bulk of greek laws and contains classified legislations from 1834 to 2015. Unfortunately, although Raptarchis archive covers a large amount of Greek legal data, little is known about its internal structure and its label organization. Despite that, as a by-product of our experiments, we managed to

extract useful relations about the label hierarchy and the inner structure of the dataset. This information was previously unknown and it was not included in the original publication of the dataset [2]. By making it available we expect to give an insight to any other NLP researcher or legal professional who shall desire to work with Raptarchis dataset in the future.

All in all, the main contributions of this dissertation are the following:

- We develop and examine different NLP techniques based on DL models and SOTA XMTC approaches in order to solve the Greek legal large-scale multi-label text classification task.
- Relied on other studies, we examine, extend and corroborate the existing bibliography and findings of the problem.
- Finally, we explore and unveil unknown information about the internal structure and the label hierarchy of the Raptarchis47k dataset.

The rest of this report is organized as follow: In Chapter 2, we give all the necessary background information about the XMTC problem and we present the most significant, in our opinion, papers in the field. Next, in Chapter 3, we study and analyze Raptarchis47k dataset from a directed acyclic graph perspective and whereas in parallel we try to extract meaningful relations about its label hierarchy and structure. In Chapter 4, we elaborate on the methods and metrics that we used for the experiments while in Chapter 5, we present our final results. Finally, in Chapter 6, we review the central concepts of this thesis and propose a blueprint of ideas for future work and applications.

2. BACKGROUND INFORMATION AND RELATED WORK

In this chapter, we provide all the necessary background material that is required for the purposes and better comprehension of this report. For this reason, in the beginning, we merely discuss the multi-label text classification problem and the parts of NLP that are related to it. Then we continue with its more extreme (XMC) version and we present some thoroughly selected papers that are in close proximity to our work. Some parts of these publications will be further explained later in Chapter 4 where we analyze the methodology and learning strategy that is followed for the experiments in Chapter 5. At this point though it is given only the general framework of the problem before we delve into its more complicated and intrigued parts.

2.1 Multi-label classification

2.1.1 General introduction to MLTC

Multilabel text classification (MLTC) or more simply text tagging is a very common research area with a wide range of real-world applications. The task is originated from the traditional text categorization problem, where each document may belong to several known topics simultaneously. Formally speaking, the problem can be viewed as the learning of a score function $f: XxY \rightarrow R$ that maps (x,y) pairs (x for the textual input and y for the corresponding label) to a score $f(x,y)$. This function have to be optimized so that it delivers large values for highly relevant text-label pairs and vice versa low scores for the uncorrelated ones. To put it more simply, this means that for any given textual input, the algorithm must assign to it the most suitable labels from a predefined set of concepts.

As a whole, MLTC is strongly related to other classification tasks. In reality, it can be seen as a generalization of the multi-class classification and at the same point as a special case of the multiclass-multioutput (also known as multitask) classification problem. In the first case, the key difference between the two is that the labels in MLTC are not mutually exclusive and it is not binding for input instances to belong to one and only one category. In contrast, there is no constraint on the number of labels that correspond to a given sample. In the

second case, the multiclass-multioutput task does not hold the binary restriction for the label properties. This means that both the number of properties and the number of classes per property are greater than two. The Fig-2.1 illustrates these key differences and the various types of the classification problem.

	Number of targets	Target cardinality
Multiclass classification	1	>2
Multilabel classification	>1	2 (0 or 1)
Multiclass-multioutput classification	>1	>2

Figure 2.1: This picture illustrates the different types of TC problem based on the target number and target cardinality [1].

2.1.2 Early solutions

In the course of time, numerous techniques have been developed to address the multi-label classification task. The early solutions commonly follow simple and rather naive approaches since they ignore critical aspects of the problem like label distribution and the scalability factor. Usually, these traditional methods can be organized into two categories [19–21]: the *Transformation methods* and the *Adapted algorithms*.

In the former case, the Transformation methods try to break down the initial problem into multiple single-label problems. These techniques benefit from the fact that a lot of work has already been made in the area of single-label classification. Hence by changing the MLTC format into an another already solved one, they can take advantage and exploit all the existing literature in that field. Some representative solutions of this type are:

1. **Transformation into binary classification problems**: This method concerns the conversion of the original MLTC into several (one per class) binary classification tasks. Binary relevance [22] and Classification chain [23] are two characteristic examples of this kind. The main idea behind these algorithms is the utilization of many binary classifiers, each per class. This implies that for any input instance, the algorithm returns as output the conjunction of all labels that have been predicted from these classifiers. The labels can be predicted independently (binary relevance) or sequentially (classification chain).
2. **Transformation into multi-class classification problem:** The algorithms of this kind construct one binary classifier for all possible label combinations. Thus, in total there are $2^{|C|}$ classifiers where $|C|$ is the number of the distinct labels in the training set. The returned output is the classifier which produce

the greatest probability. Label powerset [19] algorithms are example of this category.

3. **Ensemble methods:** These techniques use many multi-class classifiers that predict only one label class. Then the predicted outputs are combined together by an ensemble method. Such examples are the k-labelsets (RAKEL) [24, 25] and the committee machines [26].

The latter category of the traditional MLTC approaches is the Adapted algorithms. These algorithms, in contrast with the previous methods, do not require any transformation or conversion of the initial problem. Instead, they adopt various, already available, algorithms and fit them directly in the multi-label classification tasks. These techniques can be applied to many popular algorithms like k-nearest neighbors [19, 27], decision trees [19, 28, 29], and NNs [30].

2.1.3 Extreme Multi-label Classification

Extreme Multi-label Text Classification (XMTC), also known as L(arge)MTC or XMC is a special category of MLTC that, similarly to the earlier version, tags a given document with the most relevant subset of concepts from an extremely large label collection. The difference in this case lies in the fact that the number of possible classes is substantially larger and could reach hundreds, thousands, or even millions. The extreme label size, however, does not come without outgrowth consequences since it produces a plethora of new constrains and restrictions.

First of all, the enormous amount of concepts in conjunction with the data sparsity problem leads a significant portion of categories to have very few or even none (Zero-shot scenario) training instances available in the training set. This label under-representation gives birth to manifold learning obstacles since the skewed label distribution prevents the algorithms from discovering reliable dependency patterns between the categories and the input data.

Furthermore, the heavy computational cost and the scalability factor pose new limitations. For example, the traditional techniques commonly demand a considerable amount of time for training. Nevertheless, their ability to scale is rather limited since the influx of only some new labels requires the retraining of the whole model, canceling in this way all the previous learning procedure. Moreover, to add insult to injury, extra complexities come from other facets of the problem like the inner class hierarchy of the label set, the natural correlations among the concepts, the heterogeneous sources and the format of the data, etc. Finally, a prominent obstacle is the limited number of reliable XMC datasets, especially for understudied and less widespread languages like Greek.

As can be already speculated, all the traditional MLTC methods that have been presented up to now are incompetent and somehow inadequate to deal with this new version of the problem. Just to mention that transformation techniques

like label powerset require $2^{|C|}$ binary classifiers, which is prohibitive when the label set carries hundreds or thousands of concepts. Additionally, other algorithms like binary classifiers treat the classes independently, which leads mathematically to suboptimal solutions. Similar drawbacks can be detected on other non-task-specific solutions of the adapted methods (e.g. KNN, decision trees, etc).

To address the above challenges a considerable amount of work has been made and a lot of methods have been suggested in the field. These approaches generally can be split down into four categories:

1. **One-Vs-All [31–33]:** These techniques are very intuitive and reassemble to the transformation algorithms that have been discussed in the previous section. As a general rule, they treat any concept as a binary classification problem and creates a separate classifier for each label. For the reasons that we have already explained, these algorithms do not have widespread use and although they perform well on small datasets, they suffer from severe limitations.

2. **Embedding Based [34, 35]:** In contrast with the previous category, these methods confront the problem of the huge label set by compressing the labels into a low dimensional embedding space. Then, during the prediction, the outputs are decompressed again to the initial space. That process however has a serious drawback: no matter how well the compression part is designed, it always will be lost a part of the information. This makes the algorithms have only limited success.

3. **Tree-Based [34, 36, 37]:** Tree-Based algorithms try to learn a hierarchical tree structure by recursively partitioning the label space. In this manner, they can deal with the extreme number of labels in a more successful way than the 1-Vs-All and Embedding methods. Eventually, at the end of this procedure, there is only a simple classifier at each leaf with only a few active labels. As a result, the training becomes much easier and manageable.

4. **Deep Learning-Based [38–41]:** Lastly, the deep learning-based techniques are trying to take advantage of the recent developments and advancements in the area of NLP. This means that they can derive better representations from the raw text input and use them to improve the quality of their predictions. Many SOTA results in XMTC come from the methods of this category.

As it can be understood, it is impossible for someone to cover all the pertinent work and the publications that are annually released in the field of large multi-label classification. Instead, for the purposes of this thesis, we concentrate exclusively on the last category: the deep learning-based methods. For so on, all the models and techniques that we refer to will come from this specific area.

2.2 XMTC Papers

Having displayed all the background information that is necessary for the comprehension of the problem, we continue in this section with the publications that are most closely related to our work. More specifically, at this stage, we choose to present a brief synopsis of these papers whilst we will discuss them in more detail in Chapter 4 where we describe the main architecture and structure of our models.

We split this section into three separate parts. In the first place, we discuss some benchmark datasets that are representative of the task of legal large MLTC. In the second sub-section, we review those techniques which had a major impact on the the area and they have also significantly influenced the development of our models (see Section 4.1). Ultimately, the last part contains a bunch of unconventional approaches that can be employed for further study and future work.

2.2.1 XMTC Datasets

Legal Text Classifiction based on Greek Legislation [2]

Date: Dec.2020 **Datasets:** [2]

> While it is not related directly to the extreme multi-label classification task this study of Papaloukas was the predecessor of the current dissertation and thus it is useful to display it for the better comprehension of our thesis. The main contribution of this prior work is the introduction of RATPTARCHIS47k a novel dataset for multi-class/multi-label text classification on greek legislation. Furthermore, in addition to the dataset, it is tested and analyzed the performance of various techcniques, ranges from traditional ML methods to state-of-the-art transformer-based models. Some of these results will be used later in our conclusions in Section 5.2.

Large-Scale MLTC on EU legislation [17]

Date: 5 Jun. 2019. **Datasets:** [17]

> A pivotal point for the research of legal XMTC was the publication of the EURLEX57k. This dataset contains a vast number of English EU legislative documents and thus it is suitable for many multi-label classification tasks, including the cases of few-shot and zero-shot learning. In contrast with RAPTARCHIS47k, it has a more complex hierarchy (8-level instead of 3-level) as well as a greater volume of labels (7k instead of 3k). Additionally

to the dataset, the authors also investigate different baseline extreme multi-label classification methods and exhibit their results. However, most of their approaches are rather naive and do not take into account crucial factors like label hierarchy, correlations, and metadata.

LEDGAR: A Large Multi-label Corpus of Legal Provisions in Contracts [18]

Date: 11-16 May. 2020. **Datasets:** [18]

LEDGAR is another freely available dataset very similar to EURLEX. The main difference between the two is that the corpus of LEDGAR comprises solely legal provisions from contracts and official documents. For its creation, the authors elicited contracts from EDGAR (a website platform of the US Securities and Exchange Commission) and applied on them different heuristic filters to improve the quality of the data and most importantly to reduce the size of the label set. If it compared to EURLEX, LEDGAR has smaller input sentences (124 instead of 700 average tokens per sentence) but it retains a significantly larger number of labels (12k instead of 7k).

The very interesting thing however in this work is that the authors attempted to alleviate the problem of the immense label set by inferring a latent concept hierarchy. The main assumption that they did is that the labels with longer names tend to be more sparse and therefore can be considered parents of the shorter ones. So, if the long concept names are decomposed into multiple shorter classes then it is created a Directed Acyclic Graph and consequently a label hierarchy. This makes the solution of the problem much easier and simple. In the end, the authors apply different baseline models to their dataset and examine their performance and results.

2.2.2 Related Papers

As we have seen above, all the modern XMTC datasets are really challenging since they contain enormous label-sets and vast amounts of data some of which come from disparate resources.So, to ingenuously try to apply simple algorithms in such cases is quite unproductive and inefficient. Even with the contribution of state-of-the-art approaches like RNNs and Transformers, it still remains enigmatic and unclear how to improve and enhance the performance of our models.

For these reasons, in the next following paragraphs, we decided to present some brand new approaches that attempt to address the problem by using more ingenious and sophisticated strategies. Generally, these ideas include tactics like harnessing the inner concept hierarchy of the classes, uncovering hidden correlations among the labels, disassembling the original problem into simpler

ones, etc. Below we display part of these noteworthy solutions, that we have also employed in Chapter 4 for the construction of our models.

Layer-wise Guided Training for BERT [4]

Date: 12 Oct. 2020. **Datasets:** [6, 17]

First of all, a notable paper which is very close to the purposes of this work is this of Manginas et al. [2020]. In this publication, the authors cope with the XMTC problem by using only Transformers-based models. More specifically, their objective is to interconnect the predictions of different levels of the label hierarchy with specific layers of BERT. This basically means that only some carefully selected BERT layers will be responsible for the predictions of the labels on a particular hierarchical level. They believe that this process leads the models to be trained in a structured manner and therefore to accomplish better fine-tuning, parameter utilization and performance. The results or their experiments seem to confirm and verify the claims of the authors.

Correlation Networks for XMTC [42]

Date: 23-27 Aug. 2020. **Datasets:** [17, 43, 44]

While the simple deep learning models for XMTC are extremely successful in extracting information from input text sequences, most of them suffer severely when they have to extract or incorporate hidden label correlations. A very mainstream and naive approach that is employed widely in similar cases is the use of only one fully connected (FC) layer which maps a context-aware feature representation [3, 7, 9] into a new vector that represents the final predictions. However such procedures fail to capitalize and leverage the label interdependencies that exist inherently among the concepts.

In this paper, Xun et al. [2020] propose the Correlation Networks (CorNet) module, an independent computational unit that can be integrated into different ML structures (CNNs, Transformers, AttentionXML, etc.) and reinforce the raw predictions of these models with extra correlation information derived from the label set. The finding suggests that not only do these modules offer significant improvements but they also accelerate the convergence and learning rate of the whole training process.

MATCH: Metadata-Aware Text Classification in A Large Hierarchy [45]

Date: 15 Feb. 2021. **Datasets:** [46, 47]

Similar to CorNet, Zhang et al. [2021] are trying to solve the XMTC problem in a more refined way instead of using solely simple contextual information. In this case however the authors rather than relying on label correlations they exploit both the existing hierarchical structure of the label set and the available metadata that are associated with the input documents (e.g. authors' names, published venue, references, etc.). To put it another way, they concatenate together the metadata and the input word sequences so as to feed them into a Transformer model. By doing so, it is feasible for the model to capture and learn relationships between input text and metadata. Also, to boost this process, they increase the number of [CLS] tokens so that the extracted representations to become more robust and informative. Another key point in the MATCH system is that it leverages the label taxonomy. In fact, it uses various regularisation techniques to extract and obtain hypernym/hyponym (or more simply parent/children) dependencies from the label set. The final outputs indicate that the MATCH framework consistently outperforms all its state-if-the-are competitors in almost all cases and metrics.

2.2.3 Other Papers for Future Work

AttentionXML: Label Tree-based Attention-Aware DL Model for XMC [40]

Date: 25 Jun. 2020 **Datasets:** [17, 43, 48]

In contrast with all the previous methods that are based entirely on deep learning models, AttentionXML follows a more versatile approach as it combines both the prowess of DNNs with all the benefits that comes from the use of tree-based techniques. In fact, AttentionXML consists of two distinguished components. The first is a shallow and wide Probabilistic Label Tree (PLT). This allows the alogrithm to effectively handle an enormous number of concepts and even to deal "tail labels" (labels with a very few input instances).

The second component is a multi-label attention mechanism for capturing the important parts of texts which are most relevant to each category. This mechanism uses a Bi-LSTMs with attention on each layer of the tree and represents a given text differently for each label. According to the experiments, this approach can outperform other state of the art techniques in both computational time and performance while it is the best tree-based method against tail labels.

Taming Pretrained Transformers for XMTC [49]

Date: 2 Feb. 2019. **Datasets:** [48, 50–52]

A totally radical approach that sheds light on completely different aspects of the tasks is this of X-Transformers. Similar to AttentionXML, X-Transformers try to solve the initial XMTC task by decomposing it into smaller subproblems. However, instead of a tree-based schema, they employ three separate components. The first one is the Semantic Label Indexing (SLI) mechanism. This part is responsible for partitioning the vast label space into smaller label groups. To achieve that it uses ordinary clustering techniques. Next, comes the Deep Neural Matching (DNM) component that assigns the input text instances to one or more relevant clusters from the previous step. For this phase is used the power of deep pre-trained Transformers models like BERT [3], RoBERTa [53], and XLNet [54]. Finally, in the end, is following an Ensemble Ranking Method that estimates the relevance between the input text instance and the labels from the retrieved clusters. The results of the experiments indicate that XTransformer offers substantial improvements on four XMC benchmark datasets while in most cases and metrics it surpasses other state-of-the-art methods including AttentionXML.

Parameter-Efficient Transfer Learning for NLP [55]

Date: 23-27 Aug. 2020. **Datasets:** [50, 56–60]

Although the above mentioned papers address variant facets of the large multi-label classification problem, all of them suffer from a common weakness: they fail to adapt in cases when new independent tasks or datasets arrive sequentially or otherwise stated in a stream. In such scenarios, the prior algorithms retrain for any new task an entirely new model, discarding in this way not only the whole previous training effort/cost but also all the parameters that were learned beforehand. To tackle this phenomenon, the Houlsby et al. [2019] proposed the *Adapter* architecture. Adapters are bottleneck modules that can be trained incrementally to solve new -constantly arriving- downstream tasks without forgetting the previous ones. The burden of this process is only the use of a small number of new extra parameters for each new task. The findings reveal the effectiveness of the adapters as well as their ability to attain near to state-of-the-art performance while adding only a few new parameters per task.

All in all, in this chapter we go through the whole problem of large multi-label text classification. We discussed the most significant aspect of the problem and we see different approaches from the simple and traditional ones, to the most advanced and SOTA. Some of these ideas we will use later for the developments of our models later in Chapter 4, whereas others like AttentionXML, X-Transformers and Adapters convey new groundbreaking ideas for future work and study.

3. RAPTARCHIS47K DATASET

In this chapter, we introduce the dataset that we will use in our experiments later in Chapter 5. This is *RAPATARCHIS47k*, a newly released dataset proper for Greek legal text classification tasks. Besides, during the writing of this work, we have noticed that the internal label hierarchy of this dataset is not yet been fully clarified and it remains rather unclear and obscure. Therefore, we decided to enhance the literature around Raptarhcis47k by studing it as a directed acyclic graph and exploring the underlying relations/connections among the different levels of its hierarchy. To the best of our knowledge, this is the first time that the hierarchical structure of this dataset is studied and analyzed to such extent. Finally, at the end of this chapter, we give some quantitative statistics about the data that may be useful for the better understanding of our results in Section 5.2.

Our aspiration here is to offer a concrete and concise description of Raptarchis47k and at the same time to releases new insightful information about its inner structure and hierarchy. Information that can be used not only for the purposes of this thesis but also for other NLP tasks related to this dataset.

3.1 Dataset description

For all the experiments we use the *Raptarchis47k*, a novel dataset built on *Raptarchis-Permanent Greek Legislation Code (GLC)*. Raptarchis-GLC is an archive that encompasses greek statutes derived from the Official Government Gazette from the period between 1834 to 2015. The code was created by Pantelis Raptarchis and was kept initially as a private enterprise until 1978 when it was granted to the state. Currently, the code is under the surveillance of the Ministry of Interior and it is openly available to all citizens, institutions, or organizations through the e-Themis portal. In general, the code has a strict layout and follows a standard format. More precisely:

- Each article belongs to a unique Volume, Chapter, and Subject category. The volumes are ordered by number (e.g. 1,2,3 ...), the chapters by capital Greek letters(e.g. Α, Β, Γ ...), and subjects by lowercase Greek letters (e.g. α, β, γ ...).

- There is a standard three-level thematic structure with the volumes to be at the top of the hierarchy, the chapters in the second, and the subjects in the last layer. In other words, the dataset follows a depth-three level hierarchy. *Hierarchy: Volumes >Chapters >Subjects.*
- Altogether there are 47 Volume, 289 Chapter, and 2285 Subject distinct different classes.

In Fig-3.1 can be seen schematically the thematic structure of Raptarchis-GLC code.

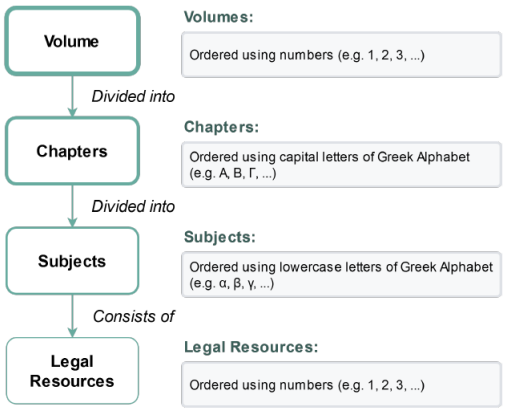


Figure 3.1: Raptarchis-GLC hierarchical structure

Lean on Raptarchis-Permanent Legislation Code, a new dataset named RAPTARCHIS47k was released in December 2020. The new dataset contains more than 47k official legislation resources allocated in TRAIN, DEV, and TEST sets. The legal resources are split into JSON files that contain the legislation (title and articles), its categories (in volume, chapter, and subject level), and other metadata (e.g. publication year, web URL, etc.).

All the properties in JSON files take string values except for the “articles” which is a list with the articles (if any available) that compose the law. Furthermore, the string values for “volume”, “chapter” and “subject” attributes indicate that any legal resource participates in one and only one category and does not belongs into omnibus bills [61] (bills that encompasses more than one irrelevant or disparate topics). This may not be true in some real-life scenarios. For example, during the Greek government-debt crisis it was introduced in the legislation corpus a number of different economic adjustment programs [62]. These programs concern bills that enclose in their scope many unrelated amendments which are though clustered together under a unique enactment. In such cases, the legal resources usually are involved in many categories simultaneously which practically means that it is required a set of classes to describe their context.

The inclusion of omnibus bills and the coverage of more recent Greek legislation (e.g. laws passed after 2015) in RAPTARCHIS47k can be delivered in future

updates of the dataset. In these next versions, the categories attributes (”Volume”, ”Chapter”, and ”Subject”) of the JSON files have to take one or more string values, all grouped in a single list that will declare the classes in which the particular law participates. Unfortunately, such instances are not available yet in this (the first) edition of dataset. However, considering that this dissertation aims to strengthen the existing studies on RAPTARCHIS47k and not to engage with the refine and production of a whole new dataset, we shall continue with this current version. So, any updates of the dataset and expansions of our models to them are left as future work since they are beyond the purview of the thesis.

Having said that, we can proceed to the next section which focuses on the analysis and examination of the current thematic label hierarchy of RAPTARCHIS47k.

3.2 Label hierarchy

During the execution of our experiments, we observed that the label hierarchy of RAPTARCHIS47k does not follow the alleged tree-based structure of Fig-3.2 that is implied on the official publication of the dataset. This fact normally would be of no significance for problems like multi-class or flat multi-label text classification since in those cases the order of the predictions does not affect the learning process. However, if we would like to develop more complex models that do not utilize only contextual information but also take into consideration the internal label hierarchy, we would like to know the real relationships that occur among the different concepts in our data.

As we have seen in Section 2.2, such techniques are more resilient and frequently offer essential improvements in performance and the explainability of the models. Moreover, in the case of legal NLP, this knowledge can be valuable for ML engineers, who want to incorporate it in their algorithms, or for other legal specialists who would desire to know the internal connections and relationships of the concepts. For all these reasons, we decided to enrich the bibliography of Raptarchis-GLC with the observations and the findings that were emerged from our experiments in Chapter 5.

3.2.1 The tree-hypothesis

As we have already mentioned, during the construction of our models we noticed that the thematic label hierarchy of the dataset does not follow the tree hypothesis that was initially expected. This hypothesis suggests RAPTARCHIS47k is a tree-based directed connected graph with no isolated points in which every node has only one parent and the vertices of the first layer (Volume level) are connected with a pseudo-node that is the root of the graph. This entails that any pair of different vertices are connected with a unique path, and the in-degree number for all vertices except the root is equal to one. Although this assumption seems very plausible and realistic for the filing of legal archives, the findings our

masking-techniques [1] reveal that Raptarchis-GLC has a more complicating and perplexing graph structure.

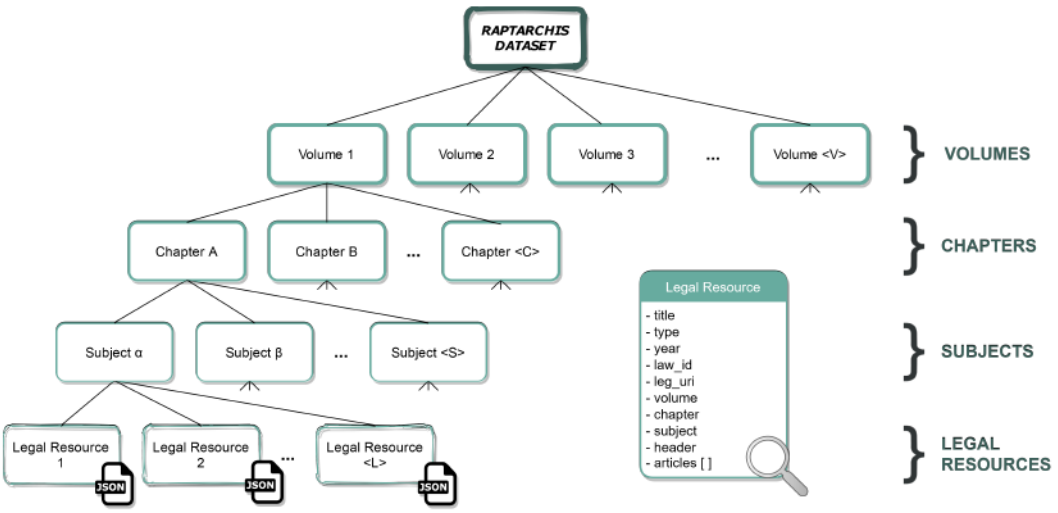


Figure 3.2: The initial tree-based assumption for Raptarchis dataset [2].

At this point, it should be clarified that in spite of the refutation of the tree hypothesis by the findings, it can become valid if we append the names of the labels with their parents' names. For example, the Subject label of *"Personnel" ("Προσωπικο")* has many Chapter parents like ***"Ministry of Justice"*** *(**"Υπουργείο Δικαιοσύνης"**)*, ***"Foreign Ministry"*** *(**"Υπουργείο Εξωτερικών"**)*, etc. If we dismiss the label name for ***"Personnel"*** and we replace it with two other *(**"Foreign Ministry> Personnel"*** and ***"Ministry of Justice> Personnel"**)* then the graph obtains a new cogent tree-based structure. Nevertheless, this tactic produces much more brand new training classes (equals to the number of parent classes for each category) and as a consequence, it reduces the total number of training instances for each category. This may be detrimental for tasks like XMTC in which we have to address the severe sparsity problem in conjunction with the extreme number of available labels. Also, in the previous example the label ***"Personnel""*** probably refers to human resources and thus it has similar context in both cases (i.e. ***"Foreign Ministry"*** and ***"Ministry of Justice"***). Consequently, it would be logically inappropriate to separate it into two different classes. For all these reasons we choose not to change the label set of the dataset and to continue with the real thematic hierarchy. Further experimentation on this topic can be implemented in future works.

[1] See Subsection 4.2.2

3.2.2 Raptarchis47k thematic Structure

To begin with, in order to unveil the hidden thematic structure of the dataset we go along with the following process: First of all, we gather all the labels that appeared in the TRAIN folder of the RAPTRARCHIS47k (label frequency > 0) and create a label graph with them. From the concepts that occur in the DEV and Test folder of the dataset, we keep only those that overlap with the labels of the training set. In general, we observe that:

- On Volume level there is 100% coverage (47/47) between the labels of training-validation and training-testing set.
- On Chapter level there is 96.37% (372/386) and 95.60% (369/386) label coverage between training-validation and training-testing set respectively.
- On Subject level there is 74.60% (1598/2143) and 74.33% (1593/2143) label coverage between training-validation and training-testing set respectively.

The extra concepts that are in the validation and testing set are considered redundant at this point since they are not appeared in the training set. However they can be extremely useful for cases of Few-show [63–65] and Zero-shot learning [63, 64, 66, 67].

Additionally, since the graph structure emerged as a consequence of the development of our masking techniques, it is not describing here how it can be used in the prediction process. A thorough explanation around these methodologies can be found later in Chapter 4. On the contrary here, we review the main attributes, properties, and relations that can be encountered in this graph. So we notice that:

1. First and foremost, It was found that many nodes have multiple parents or indegree> 2. As a matter of fact, there were detected 3 labels in the Chapter layer with indegree value equal to two and 25 labels on the Subject layer with indegree value greater or equal to two.
2. There are labels in the dataset that point down to another label with the same name. From the graph's perspective, this similarity is only nominal and the parent node does not force the child node to have the same children with it (no recursive relationship).
3. It can be found labels in the different levels of the graph that have identical names. Nonetheless, unlike the previous case, these vertices are located on different branches/locations of the graph and do not belong on the same path (no ascendant/descendant association).

In the Figure 3.3 we illustrate some of the above cases with real data from Raptarchis dataset. So, for example, we observe that the Subject label ***"ΤΑΜΕΙΑΚΗ ΥΠΗΡΕΣΙΑ"*** has at least two parents (***"ΟΙΚΟΝΟΜΙΚΗ ΔΙΟΙΚΗΣΗ"*** and ***"ΕΙΣΠΡΑΞΗ ΔΗΜΟΣΙΩΝ ΕΣΟΔΩΝ"***). Also, the label ***"ΟΙΚΟΝΟΜΙΚΗ ΔΙΟΙΚΗΣΗ"*** appears in both Volume and Chapter level without though to belong

in the same path (3rd case). Finally, the label ***"ΔΙΑΦΟΡΟΙ ΝΟΜΟΙ"*** has many incoming edges (indegree value equal to 5).

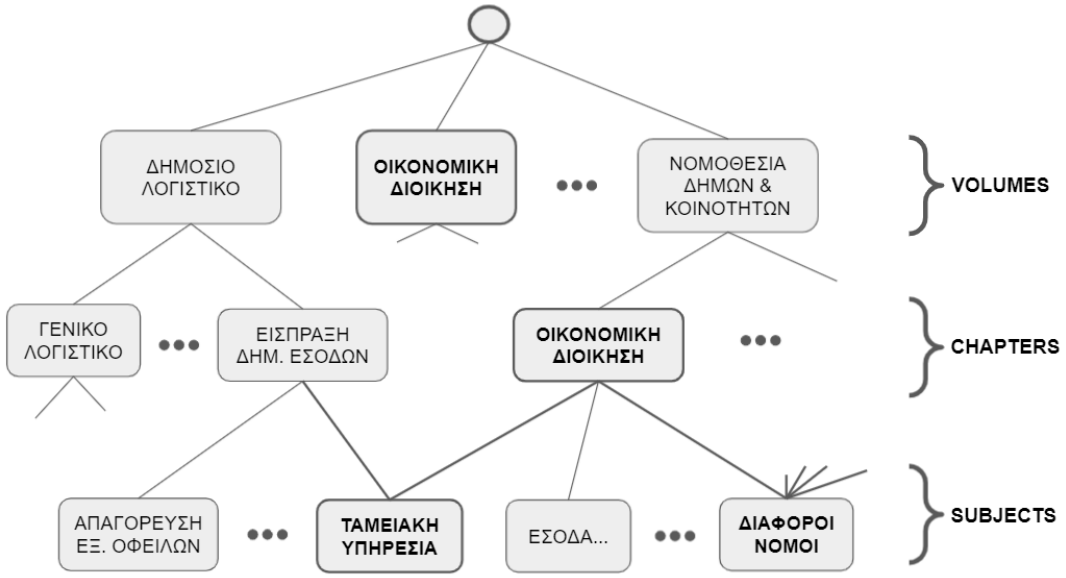


Figure 3.3: A real instance of RAPTARCHIS47k that depicts the DAG structure of the dataset.

A more elaborate and meticulous presentation of the above results is given on the Github folder of the thesis. There, we maintain four txt files which contain a detailed description of the outputs that was arisen from this chapter. For example, there are presented all the labels with multiple parents, the nodes which share the same name, and other additional information like dataset description, etc. Finally, some of those conclusions are included in the Appendix at the end of this report.

3.3 Data Analysis

Having given a detailed description of the RAPTARCHIS47k dataset and its inner label hierarchy, we proceed this study with a brief investigation and analysis of our data.

To begin with, all the concepts in the dataset can be divided into three categories based on their appearance and population frequency. Hence, there are the frequent concepts that include the labels that have more training instances than a predefined threshold, then the few-shot or low frequent concepts that have training instances lower than this threshold (but not zero) and lastly, the zero-shot classes that appear only in the development and test set. Here, we are interested in the first category and therefore we engage only with the frequent labels of the dataset.

Table 3.1: The size of Raptarchis47k dataset for two different Frequency Thresholds.

	Freq-Num:10	Freq-Num:50
Training Set	23412	9520
Testing Set	7604	3093
Validation Set	7615	3062

For our experiments, we set the two default threshold with frequent numbers equal to 10 and 50. This means that we keep only the concepts that occur at least 10 (or 50) times in the training set. All the other classes are dismissed and do not participate in the training process. The table Table 3.1 shows the size of Raptarchis47k dataset (Train, Valid. and Test set) after the removal of the rare labels. Furthermore, on the Table 3.5 are displayed the categories that appeared at least 10 times in the Training set while in parenthesis is included the corresponding cardinality number of each label. The Table 3.6 is equivalent but with Threshold value equal to 50.

Table 3.2: The total label distribution per category for two different Frequency Threshold.

	Freq-Num:10	Freq-Num:50
Volumes	47	39
Chapters	300	113
Subject	767	111

At this moment, it is essential to mention how the training set is constructed and how we decide which instances will be selected and which ones will be discarded. First, we gather all the samples together in a single collection (dataframe) and we dismiss those which have Volume-Frequency lower than 10 (or 50). Next, in the new collection, we dismiss all the samples which have Chapter-Frequency lower than 10 (or 50) and so on for the Subject-Level. In this way, we ensure that our samples will be consistent and there will not exist for example instances that are marked with a prevailing label (high frequency) in the Volume layer but with a rare one (occurrences less than the Threshold) in Chapter or Subject layer. This method however may create some discrepancies in cardinalities among the items of Table 3.5 and Table 3.6. This is the reason why the labels and the label order between the two tables it is not identical but differs even for the most prevailing concepts.

Going forward, the Table 3.3 shows the average length (by tokens) of the legal documents in the dataset. As it is already known each document consists of a header and a body which is the concatenation of its article list. For the training, we choose to take advantage of both the header and the body of the legal resources and thus we concatenate them into a unified textual sequence. Then, we set the maximum length of BERT input equals to 400 tokens. Documents bigger than

Table 3.3: Average tokens per Header and Articles

		TRAIN	VALIDATION	TEST
Avg. Token FreqNum-10	**Header**	113.16	119.98	115.59
	Articles	442.87	387.42	421.26
Avg. Tokens FreqNum-50	**Header**	110.71	114.26	101.61
	Articles	374.18	338.53	408.64

Raptarchis's Dataset

this limit are truncated, while those which are smaller are padding with zeros. Thankfully though as can be seen in the Table 3.3 most of the information is preserved. More information about the dataset and the data distribution can be found in Appendix.

Table 3.4: A Real instance of Raptarchis Permanent-GLC label structure

RAPTARCHIS GLC		
Volumes	**Chapters**	**Subjects**
ΓΕΩΡΓΙΚΗ ΝΟΜΟΘΕΣΙΑ	ΔΙΑΦΟΡΑ	ΚΑΝΝΑΒΙΣ
		ΓΕΩΜΗΛΑ
		ΜΑΣΤΙΧΑ
		ΔΙΑΦΟΡΑ
		...
	ΥΔΑΤΑ	ΑΡΔΕΥΣΕΙΣ
		ΓΕΩΤΡΗΣΕΙΣ
		ΒΙΟΤΟΠΟΙ ΚΑΙ ΟΙΚΟΣΥΣΤΗΜΑΤΑ
		...
	...	...
ΠΟΙΝΙΚΗ ΝΟΜΟΘΕΣΙΑ	ΕΙΔΙΚΟΙ ΠΟΙΝΙΚΟΙ ΝΟΜΟΙ	ΛΗΣΤΕΙΑ
		ΠΡΟΔΟΣΙΑ
		ΦΥΓΟΔΙΚΙΑ
		...
	ΔΙΕΘΝΕΣ ΠΟΙΝΙΚΟ ΔΙΚΑΙΟ	ΔΟΥΛΕΙΑ
		ΓΕΝΟΚΤΟΝΙΑ
		ΤΡΟΜΟΚΡΑΤΙΑ
		...
	ΑΕΡΟΠΟΡΙΚΟ ΠΟΙΝΙΚΟ ΔΙΚΑΙΟ	ΑΕΡΟΔΙΚΕΙΑ
	...	...
ΕΘΝΙΚΗ ΑΜΥΝΑ	ΣΤΡΑΤΟΛΟΓΙΑ	ΕΦΕΔΡΟΙ
		ΑΝΥΠΟΤΑΚΤΟΙ
		ΑΝΑΒΟΛΗ ΣΤΡ.
		ΣΤΡ. ΓΥΝΑΙΚΩΝ
		...
...	...	...

Raptarchis's thematic structure

Table 3.5: The label tags of Raptarchis47k that have at least 10 instances in the Training set.

Frequency-Threshold: 10	
Volumes	ΕΠΙΣΤΗΜΕΣ ΚΑΙ ΤΕΧΝΕΣ (1779), ΕΚΠΑΙΔΕΥΤΙΚΗ ΝΟΜΟΘΕΣΙΑ (1595), ΣΥΓΚΟΙΝΩΝΙΕΣ (1294), ΓΕΩΡΓΙΚΗ ΝΟΜΟΘΕΣΙΑ (990), ΕΜΠΟΡΙΚΗ ΝΑΥΤΙΛΙΑ (919) **...** ΠΟΛΕΜΙΚΟ ΝΑΥΤΙΚΟ (181), ΠΟΛΙΤΙΚΗ ΑΕΡΟΠΟΡΙΑ (174), ΑΓΡΟΤΙΚΗ ΝΟΜΟΘΕΣΙΑ (155), ΠΟΛΕΜΙΚΗ ΑΕΡΟΠΟΡΙΑ (140), ΠΟΛΙΤΙΚΗ ΔΙΚΟΝΟΜΙΑ (84),
Chapters	ΑΥΤΟΚΙΝΗΤΑ (742), ΔΙΑΦΟΡΑ ΑΣΦΑΛΙΣΤΙΚΑ ΤΑΜΕΙΑ (714), ΥΠΟΥΡΓΕΙΟ ΕΞΩΤΕΡΙΚΩΝ (401), ΕΠΑΓΓΕΛΜΑΤΙΚΗ ΕΚΠΑΙΔΕΥΣΗ (335), ΦΟΡΟΛΟΓΙΑ ΚΑΘΑΡΑΣ ΠΡΟΣΟΔΟΥ (330), **...** ΕΞΟΔΑ ΠΟΙΝΙΚΗΣ ΔΙΑΔΙΚΑΣΙΑΣ (11), ΥΓΕΙΟΝΟΜΙΚΗ ΥΠΗΡΕΣΙΑ ΣΤΡΑΤΟΥ (11), ΕΚΠΑΙΔΕΥΣΗ ΕΡΓΑΤΩΝ (10), ΜΑΘΗΤΙΚΗ ΠΡΟΝΟΙΑ (10), ΑΠΟΣΤΟΛΙΚΗ ΔΙΑΚΟΝΙΑ ΕΚΚΛΗΣΙΑΣ ΤΗΣ ΕΛΛΑΔΟΣ (10)
Subjects	ΓΕΝΙΚΕΣ ΔΙΑΤΑΞΕΙΣ (776),ΟΡΓΑΝΙΚΕΣ ΔΙΑΤΑΞΕΙΣ (375), ΓΕΝΙΚΑ ΠΕΡΙ ΚΥΚΛΟΦΟΡΙΑΣ ΑΥΤΟΚΙΝΗΤΩΝ (186), ΠΡΕΣΒΕΙΕΣ ΚΑΙ ΠΡΟΞΕΝΕΙΑ (186), ΟΡΓΑΝΙΣΜΟΣ ΥΠΟΥΡΓΕΙΟΥ ΓΕΩΡΓΙΑΣ (148), **...** ΟΡΓΑΝΙΚΕΣ ΔΙΑΤΑΞΕΙΣ ΑΠΟΣΤΟΛΙΚΗΣ ΔΙΑΚΟΝΙΑΣ (10), ΣΥΜΦΩΝΙΕΣ ΠΡΟΣΤΑΣΙΑΣ ΤΟΥ ΠΕΡΙΒΑΛΛΟΝΤΟΣ (10), ΗΘΙΚΕΣ ΑΜΟΙΒΕΣ ΠΡΟΣΩΠΙΚΟΥ (΄ΕΝΟΠΛΟΥ-ΠΟΛΙΤΙΚΟΥ) ΥΠΟΥΡΓΕΙΟΥ ΔΗΜΟΣΙΑΣ ΤΑΞΗΣ (10), ΑΝΑΓΝΩΡΙΣΗ ΞΕΝΩΝ ΚΑΤΑΜΕΤΡΗΣΕΩΝ (10), ΑΝΑΣΤΟΛΕΣ ΤΟΥ ΣΥΝΤΑΓΜΑΤΟΣ - ΚΑΤΑΣΤΑΣΗ ΠΟΛΙΟΡΚΙΑΣ (10)

Raptarchis's Label Frequency-10

Table 3.6: Similar to the Table 3.5 here there are the labels of Raptarchis47k that have at least 50 instances in the Training set.

Freq-Num: 50	
Volumes	ΣΥΓΚΟΙΝΩΝΙΕΣ (990), ΕΠΙΣΤΗΜΕΣ ΚΑΙ ΤΕΧΝΕΣ (770), ΕΚΠΑΙΔΕΥΤΙΚΗ ΝΟΜΟΘΕΣΙΑ (638), ΓΕΩΡΓΙΚΗ ΝΟΜΟΘΕΣΙΑ (546), ΔΙΟΙΚΗΣΗ ΔΙΚΑΙΟΣΥΝΗΣ (458), **...** ΣΤΡΑΤΟΣ ΞΗΡΑΣ (54), ΝΟΜΟΘΕΣΙΑ ΔΗΜΩΝ ΚΑΙ ΚΟΙΝΟΤΗΤΩΝ (54), ΠΕΡΙΟΥΣΙΑ ΔΗΜΟΣΙΟΥ ΚΑΙ ΝΟΜΙΣΜΑ (51), ΑΣΦΑΛΙΣΤΙΚΑ ΤΑΜΕΙΑ (51), ΠΟΛΕΜΙΚΗ ΑΕΡΟΠΟΡΙΑ (15)
Chapters	ΑΥΤΟΚΙΝΗΤΑ (622), ΥΠΟΥΡΓΕΙΟ ΕΞΩΤΕΡΙΚΩΝ (344), ΦΟΡΟΛΟΓΙΑ ΚΑΘΑΡΑΣ ΠΡΟΣΟΔΟΥ (299), ΠΑΝΕΠΙΣΤΗΜΙΟ ΙΩΑΝΝΙΝΩΝ (236), ΥΠΟΥΡΓΕΙΟ ΓΕΩΡΓΙΑΣ (208), **...** ΣΙΔΗΡΟΔΡΟΜΟΙ ΓΕΝΙΚΑ (6), ΥΔΡΕΥΣΗ (6), ΔΗΜΟΣΙΟ ΧΡΕΟΣ (5), ΙΔΙΩΤΙΚΟ ΝΑΥΤΙΚΟ ΔΙΚΑΙΟ (2), ΟΡΓΑΝΩΣΗ ΧΡΟΝΟΥ ΕΡΓΑΣΙΑΣ (1)
Subjects	ΓΕΝΙΚΕΣ ΔΙΑΤΑΞΕΙΣ (621),ΟΡΓΑΝΙΚΕΣ ΔΙΑΤΑΞΕΙΣ (338), ΓΕΝΙΚΑ ΠΕΡΙ ΚΥΚΛΟΦΟΡΙΑΣ ΑΥΤΟΚΙΝΗΤΩΝ (186), ΠΡΕΣΒΕΙΕΣ ΚΑΙ ΠΡΟΞΕΝΕΙΑ (186), ΟΡΓΑΝΙΣΜΟΣ ΥΠΟΥΡΓΕΙΟΥ ΓΕΩΡΓΙΑΣ (148), **...** ΕΠΙΒΑΤΗΓΑ ΜΕΣΟΓΕΙΑΚΑ ΚΑΙ ΤΟΥΡΙΣΤΙΚΑ ΠΛΟΙΑ (50), ΑΞΙΩΜΑΤΙΚΟΙ ΚΑΙ ΥΠΑΞΙΩΜΑΤΙΚΟΙ Λ.Σ (50), ΒΟΥΛΗ (50), ΠΟΛΕΜΙΚΗ ΔΙΑΘΕΣΙΜΟΤΗΤΑ (50), ΣΥΜΒΟΥΛΙΟ ΕΠΙΚΡΑΤΕΙΑΣ (50)

Raptarchis's Label Frequency-50

4. METHODOLOGY

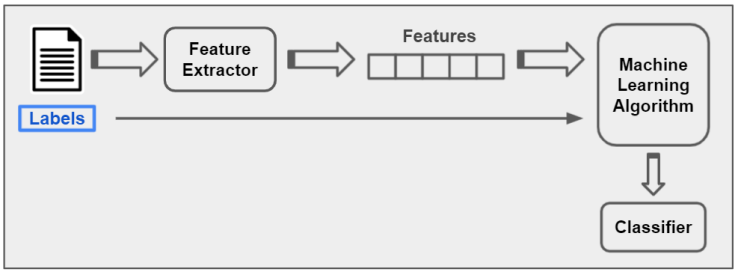
We start this chapter by giving an overview of the general process that is employed for the development and training of our algorithms. This methodology is very common in AI and it is used universally amongst the various classification problems. Next and before proceeding with the demonstration of our algorithms we focus on the BERT LM. BERT is a significant component in the structure of our models and so we dedicate a brief part of this chapter to display and analyze it. After that, we continue with the presentation of our methods (most of them originate from the papers that were reviewed in Section 2.2). Finally, in the last section, we go through the evaluation measurements that will be used for the appraisal of our models later in Chapter 5.

4.1 Text Representation

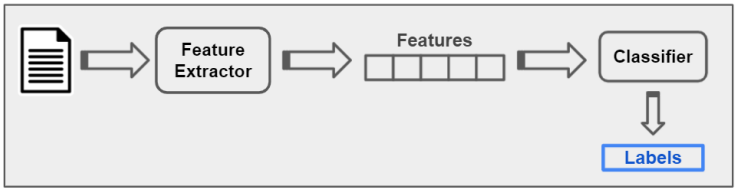
4.1.1 Traditional Approaches

In most text classification problems, the models, instead of relying on manually crafted human-made rules, learn to automatically improve their performance by using past experience and observations. In fact, they leverage the great amount of data that are nowadays available, in order to discover meaningful relations among the textual inputs and the desired outputs. Usually, the standard strategy towards training an ML classifier involves the preprocessing/cleaning of the input data, a feature extraction method that elicits useful information from the text sequences, and finally a classification component that converts the extracted features into label probabilities. The figure Figure 4.1 illustrates this procedure.

One critical factor for the successful completion of the training task is the usage of a proper text representation method. These methods are responsible for the conversion of the unstructured text sequences into new expressive numerical vectors that the machines can understand and perform mathematical computations with them. That denotes that if the new transformed representation does not capture rich semantic contextual information that lies in the input data, or if the knowledge that conveys is very poor then the classifier will not be able to accomplish the task successfully. To deal with that issue a lot of techniques and



(a) Training process



(b) Prediction phase

Figure 4.1: The standard workflow of a Machine Learning algorithm.

bibliography have been proposed over time.

The most noteworthy of these suggestions are :

1. **Bag of words (BoW)**: A very common text modeling technique that uses multisets (bags) of tokens in order to encode the occurrences of words in a document is the BoW. In practice, it is a very simple and effective method, especially when it is used in conjunction with other complementary algorithms like TF-IDF. In this case, for example, it can improve its performance by discarding all the commonly used terms and select only the most significant and informative ones. Nonetheless, despite its simplicity, this method still suffers from severe limitations as it ignores crucial factors like word order, contextual information, and semantic similarity between the different tokens in the sentences.

2. **Word embeddings [68, 69]**: A more sophisticated and elegant family of techniques for producing word representations is the word embeddings. Word embeddings are low-dimensional numerical vectors that are trained on vast corpora of unlabeled data. The main advantage of these vectors is that, in contrast with BoW, they can encode the word meanings, semantics, and to some extent the underlying syntax and grammar in the text. This practically means that similar words will have similar representations and thus, there will have close spatial proximity in the vector space. Nevertheless, these techniques still hold vital shortcomings as their produced vectors are fixed/static and hence are independent of the input context. Furthermore, they cannot confront the polysemy/homonymy problem. Popular pre-trained word embedding collections are: Word2Vec [69], FastText [70], and Glove embeddings [71].

3. **Recurrent Neural Networks (RNN) [8, 9, 72]**: RNNs variants like Long

Short-Term Memory (LSTM) and Gated Recurrent Units (GRU) are special mechanisms that apt enough to tackle many of the previous problems. These methods do not depend only on a single data point, but they utilize the entire input sequences. Moreover, some version like LSTM holds memory cells so as to retain and preserve information for a long period of time. This allows them to capture long-term dependencies and therefore to produce not static but contextually meaningful embeddings that are relevant to the input context. On the downside, these networks are computationally expensive, overfit easily, and in practice, they cannot handle very long sequences.

4. **Other:** Novel approaches in the area are the Embeddings from Language Model (ELMo) [73], Universal Language Model Fine-tuning (ULMFiT) [74], OpenAI Generative Pre-trained Transformer 3 (GPT-3) [75], and more recently the Transformers [10] and the Bidirectional Encoder Representations from Transformers (BERT) [3]. With this last family of methods, we will be occupied extensively in the next few paragraphs.

4.1.2 BERT component

During the last past years, the advent of Transformers and transformer-based models like BERT transformed the landscape and the research in the field of NLP. The outstanding performance of BERT and its successors (RoBERTa, XLNet, etc) in a plethora of different problems and their ability to adapt to many downstream tasks led to the replacement of the previous SOTA methods like LSTMs and GRUs by these novel techniques. Nowadays the BERT-Based models are the default approach in many NLP problems like QA, NER, TC, etc.

Broadly speaking, BERT LM is an open-source framework that is pre-trained with vast amounts of unlabeled data (including English Wikipedia) on the tasks of MLM and NSP. These tasks are suitable for Bidirectional training since they allow the read and process of the entire input text sequence at once. However, it would be more accurate to say that BERT is a non-directional model since its attention mechanism does not treat the input with the traditional right-to-left or left-to-right ways of Bi-LSTMs and Bi-GRUs. At the end of the day, however, the ultimate purpose of the pre-training process is the enhancement of the model's ability to understand textual contexts and to deal with complex language ambiguities.

After that procedure, BERT can be fine-tuned on numerous downstream problems (e.g. QA, NER, TC) with the use of other smaller task-oriented datasets. At this phase, the model learns to adjust to the new dataset while in general, keeps the pre-trained weights unchained. As a result, all the fine-tuned models have distinct weights, even though they are initialized with the same pre-trained parameters. The Figure 4.2 illustrates the *pre-trained* and *fine-tuning* functions of the framework.

The effectiveness of this structure on English corpora led to its extension to other

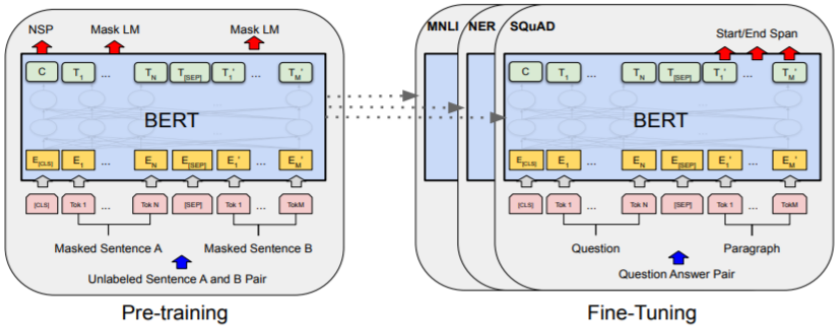
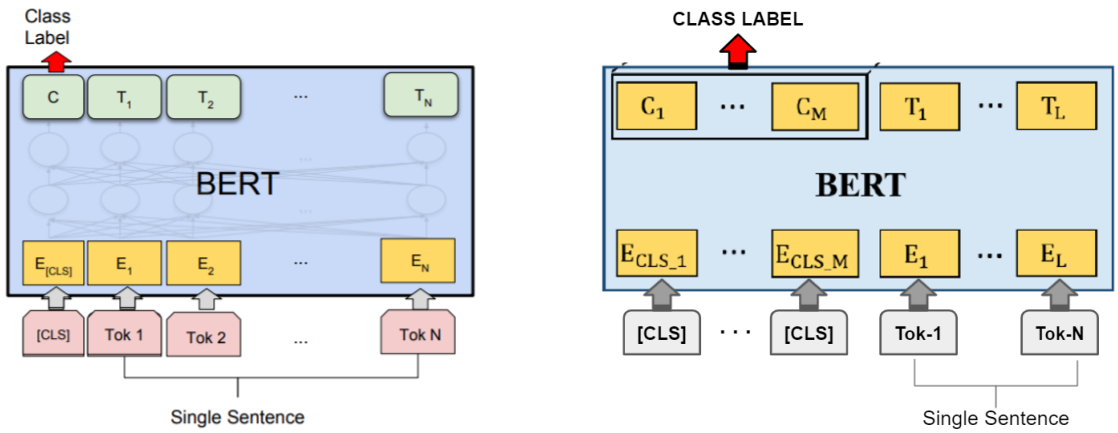


Figure 4.2: The two main procedures of BERT are the *Pre-training* and the *Fine-Tuning* functions [3].

languages either with the use of multilingual or monolingual approaches. For Greek, there are many multilingual models like M-BERT [3] and XLM [76], with the first greek monolingual version, the Greek-BERT [77], to be released in Sept. 2020. The main difference between the two methodologies is that multilingual models deploy only a small portion of their vocabulary for Greek (1%-2%) while the monolingual versions are trained exclusively on Greek documents and therefore can be performed better on similar datasets. Indeed, The experiments [77] indicate that although the models produce similar results in many tasks (e.g. NLI, PoS, and NER) the Greek-BERT perform better in most of the cases. In the meantime, both approaches surpass all the other competitors (Bi-LSTM, CNN, and word embeddings).

The Figure 4.3 illustrates how the BERT structure can be used on tasks like single sentence or text classification. Normally, for sequence-level problems, it is exploited only the classification token of the last layer of BERT in order to aggregate the sequence representation for the downstream classification task (Fig- 4.3a). However, sometimes the use of a single [CLS] token is not sufficient enough to capture the whole information of the input text. In such scenarios, a common practice is the addition of extra [CLS] tokens in front of the input sequence (Fig- 4.3b). Later in our experiments, we will utilize both these practices.



(a) Unique-CLS-token **(b) Multiple-CLS-tokens**

Figure 4.3: The usage of BERT module for the Text Classification problem.

4.2 Models

As it has been explained before, the Transformer-based techniques hold significant advantages in comparison with their other competitors. Moreover, for the Raptarchis dataset, we already have strong indicators [2] that even simple BERT approaches are capable to yield the best and finest results for the task of multi-class Text Classification. Based on this a priori knowledge, we decided to concentrate our attention exclusively on BERT-Based techniques in order to solve the Greek legal multi-label classification problem.

In the next few paragraphs, we showcase all the models that were designed during the writing of our thesis as well as the interpretation and reasoning behind them. For simplicity, we partitioned our methods into three main categories based on their composition and congenital properties.

4.2.1 Layer-wise Guided methods

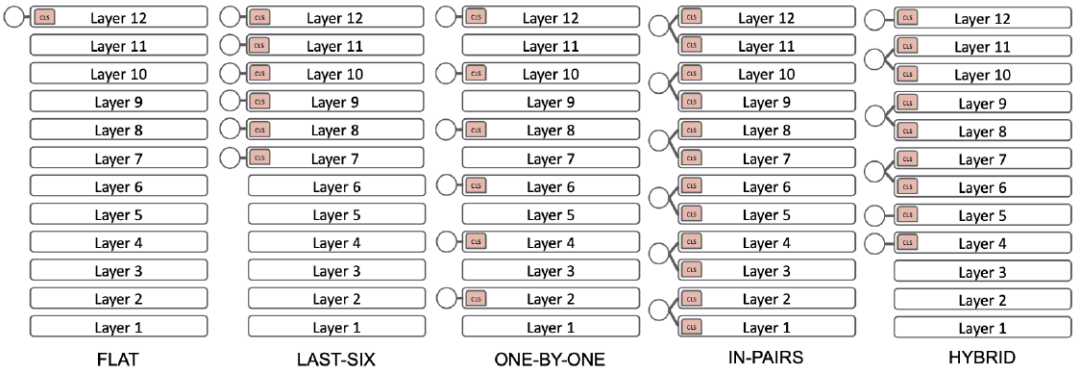
Our first attempt towards dealing with the Greek legal XMC is based on the paper of Manginas et al. [2020] [1]. In this work, the authors propose a layer-wise structured training for BERT-Based methods, that aims to solve the MLTC task in a hierarchical style. In practice, they lead different parts of BERT to successively predict the concepts that belong to a particular level of the label graph. This has, as a result, the initial problem to be dismantled into many smaller classification or multi-label classification sub-problems of increasing complexity.

A decisive factor in the development of these models is the label hierarchy of the dataset. In the original publication, the hierarchal structures of the datasets were refined to cover only the first 6 levels of the actual graph, while the other layers are truncated and discarded as redundant. After that process, the authors continued with the development of five different approaches (Fig- 4.4a) for guided training, that are specifically customized to the modified label hierarchies.

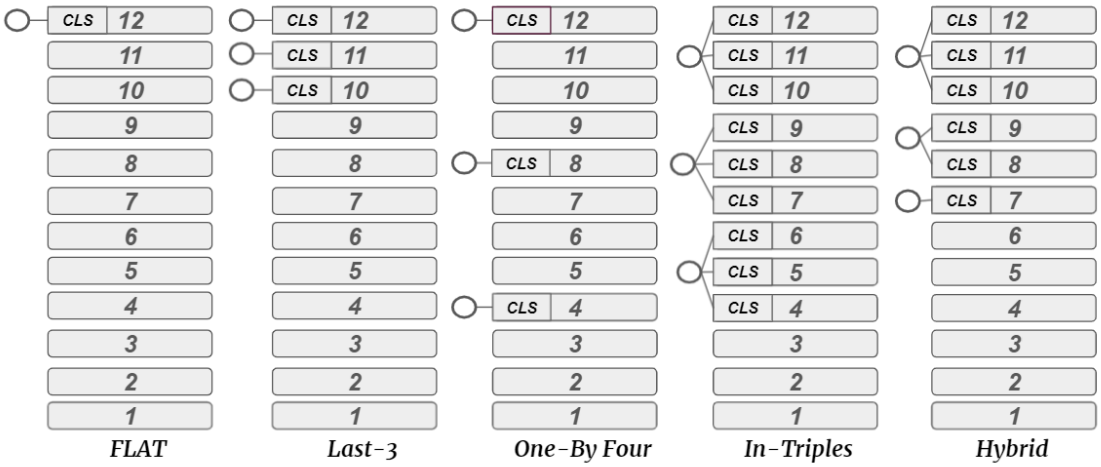
In this subsection, we demonstrate the first group of our algorithms that are a generalization of the above techniques to the necessities and demands of the RAPTRCHIS47k dataset. The central complication here is how the new models will be redesigned so as to pander to the requirements of the depth-3 hierarchy of Raptarchis's label DAG. Additionally, another key role in this reconstruction has the total number of labels (cardinality) on each level of the graph (Table 3.2). Below we deliver an overall description of these methods:

1. **SimpleBERT** It is the baseline and most simple form of our techniques. This approach utilizes as the feature vector of the input text representation the [CLS] token of the last layer of BERT (Fig- 4.3a). This vector is used by three

[1] See also Section 2.2



(a) The exact structure of the initial models.



(b) The adaptation of the above models for RAPTARCHIS47k dataset.

Figure 4.4: The presentation of the original models of *Magninas et al (2020)* [4] and their adjusted versions for the Raptarchis-GLC.

discrete classifiers that each one corresponds to one of the three hierarchical levels of Raptarchis hierarchy (i.e. Volume, Chapter, and Subject). The classifiers are responsible for predicting the classes (unique labels) for the level that they represent. Generally speaking, this method differs from the flat MLTC since the labels are already separated by layer and they are not assembled together in a single unsorted collection.

2. **Last3:** In contrast with SimpleBERT that employs only one BERT layer for three levels of predictions, Last3 utilizes the last three [CLS] tokens of the BERT. Hence, Layer-10 ($L10$) is used for the Volumes, $L11$ for the Chapters, and $L12$ for the Subjects predictions respectively. The intuition here is that we allow layers 1 to 9 to maintain their pre-trained functionality, whereas layers 10-12 will be leveraged to solve three simple classification sub-problems. It is expected that this strategy will lead to higher parameter utilization for layers 1-9 since they gradually learn to address more refined and challenging TC tasks.

3. **One-by-Four:** This method spread the classification task across the whole depth of the BERT model in an attempt to manage better parameter utilization. In essence, it adopts a *skip-3, use-1* approach in which the layers

$L4, L8$, and $L12$ (or $Li : i\%4 == 0$) are committed with the classification part while the others continue to hold their previous functionality. Nevertheless, a major risk of this method is that it will likely distract the model's pre-trained functionality and thus it will deteriorate the overall performance.

4. **In-Triples:** This approach tries to exploit the full depth of the BERT architecture in combination with the expressive power that comes from the various [CLS] tokens on different layers. To this purpose, it splits the 12 layers of BERT into four trinities. The first (L1-3) remains intact and does not participate in the classification process. The second (L4-6) is charged with the task of Volume classification. This means that it concatenates the three [CLS] tokens of these layers in order to create a unified and more robust feature vector that is used to predict the volume category of a given input. In the same way, the third trinity is responsible for the Chapter's, and the last for the Subject's predictions.

5. **Hydrid:** Although In-Triples method enhances the feature representation with many [CLS] tokens from different layers, it would be rather naive to use it without taking into account the real number of the concepts on each hierarchical level. For instance, if the second layer of the label graph contained 500 concepts while the first had only 10, then it would be careless to allocate evenly the same number of [CLS] tokens to both. The Hybrid model tries to fix this issue by giving more expressive power to the hierarchy levels that are encumbered with an intense population of concepts while in the meantime restricts the sparse ones. For RAPTARCHIS47k this means that for FreqNum-10 scenario (see Table 3.1) the structure will resemble the last model of Fig- 4.4b.

The core hypothesis behind these models is that the structured training of Deep Transfomer-Based models like BERT can lead not only to finer results but also to better fine-tuning, and parameter utilization. Furthermore, as we will see later in Section 5.2 such techniques can disclose insightful results about the inner functionality and the explainability of BERT component. The visual depiction of the above approaches in comparison with their original predecessor are displayed on Figure 4.4

4.2.2 Masking techniques

As we have seen earlier all the above models guide specific layers of BERT to sequentially predict the categories for certain hierarchical levels. Therefore, firstly are produced the probabilities for the Volumes, next for the Chapters, and lastly for the Subjects. This implies that the predictions are produced independently and do not influence each other to the final output.

In this part, in order to tackle this phenomenon, we expand the previous methodology by introducing our so-called masking techniques. These techniques exploit the outcomes of the classes that are higher in label hierarchy to improve

the predictions of the lower levels. The construction of such models is intuitively reasonable given that the production of the outputs occurs in succession (from the higher to lower levels: $L1 \Rightarrow L2 \Rightarrow L3$) and the final results are logically interrelated with each other.

To achieve our goals, first and foremost, we construct two affinity matrices that encode the interconnections among the classes on the label graph. Every row in the matrices expresses the concepts of a certain layer $Li : i \in \{1, 2\}$ and the columns represent the next $L_{i+1} : i \in \{2, 3\}$ layer. All the elements of the matrix take solely zero/one values, with the one to signify that there is a parent-child relationship between a particular row-column pair, while the zero elements that the two nodes are unrelated (no edge between them). The two matrices are the following:

1. **Vol2Ch:** A 2D array of $|VolumesNum| * |ChaptersNum|$ dimensionality that describes the Volume-Chapter relations in Raptarchis-GLC.
2. **Ch2Sub:** A 2D array of $|ChaptersNum| * |SubjectsNum|$ dimensionality that represents the Chapter-Subejct relations in Raptarchis-GLC.

Before we proceed with the methodology, it is noteworthy to mention that if we sum -for instance- the *Vol2Ch* matrix by rows, then we take a $1x|VolumesNum|$ vector that contains the number of all Chapter-children that correspond to every Volume. Respectively, if we sum it by columns then we have a $1x|ChaptersNum|$ vector that exhibits how many Volume-parents has each Chapter node. From these observations, we deduced the findings of the thematic label structure and hierarchy of Raptarchis-GLC that was presented in Section 3.2.

A real-world example is demonstrated in Figure 4.5. In this picture, we see the Chapter-to-Subject (*Ch2Sub*) matrix for the FreqNum50 scenario. The dimensions of the tensor are $113 * 111$ with the rows to stand for the chapter and columns for the subject labels. If we sum it by rows, then we take the $(1 * 113)$ vector that is in the second paragraph of the image. This vector shows the total children (outdegree value) for every chapter node. Similarly, if we sum the tensor by columns, then we have the $(1 * 111)$ vector of the third paragraph of the image which shows how many parents (indegree value) has each subject node. If the graph was a tree then all the elements of this vector should be one. However, this is not the case for RAPTARCHIS47k dataset.

As noted, these matrices express the concept relations that exist among the different levels of label hierarchy. By including this knowledge into our algorithms we add extra untapped information to the training process. This incorporation can be accomplished in many different ways. Here, we suggest a simple and intuitive technique that can be integrated easily not only in the models of the Figure 4.4 but also to any other algorithm with similar architecture. The methodology is the following:

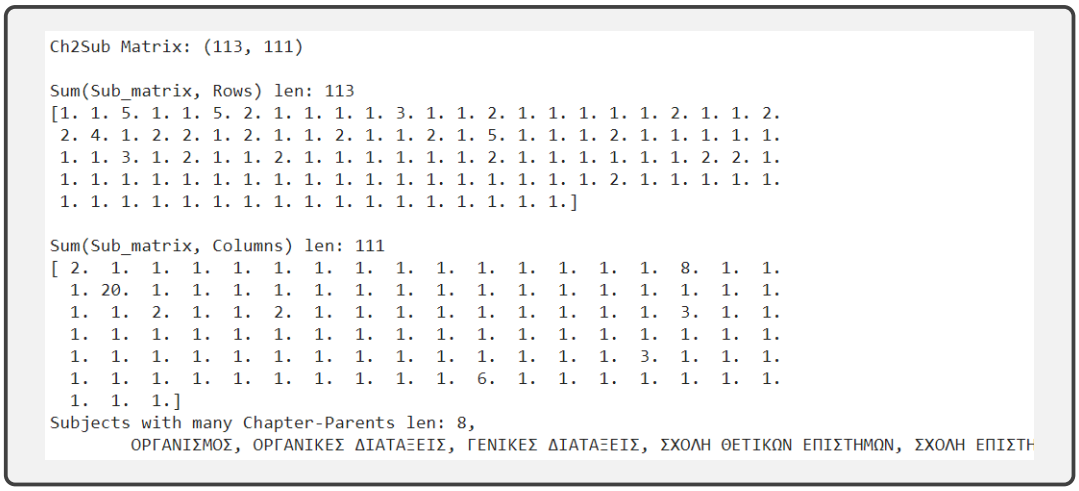


Figure 4.5: The sum by Rows and Columns of the *Chapter-to-Subject* matrix for FreqNum50 scenario.

1. First we produce sequentially the raw predictions for every layer of the label graph.[2] For RAPTARCHIS47k we dub these outputs as *vol-output, ch-output, and sub-output*.
2. Next, we calculate the dot product of *vol-output*[3] and *Vol2Ch* matrix in order to produce a *Mask* vector. This step is depicted in Figure 4.6.
3. Finally, we yield the new enhanced *ch-output* vector, by multiplying elementwise [4] (Hadamard product) the old *ch-output* predictions with the *Mask* of the previous step.
4. We repeat the same process on the next (Subject) level.

The whole process is presented algorithmically on Algorithm-1 and schematically in Figure 4.7.

Overall,the above methodology is established in two main premises. The first is that we can achieve better performance for the categories that are higher in the label hierarchy. This is usually true since the concepts that are on the upper layers of the label graph are typically fewer and more sparse. So, the MLTC task is more manageable and easy to be solved. The second premise is that already exists or can be extracted a reliable label structure for the concepts. If this structure is given, then it can be utilized directly or with a few amendments like those of [4]. If it is not provided then still there are techniques like this of [18] (see Section 2.2) that are capable to extract a well-founded and solid thematic label hierarchy. For RAPTARCHIS47k dataset, however, both assumptions are valid and therefore it is permissible the application of this algorithm.

[2]For the generation of the raw predictions we use the BERT-Based techniques of the previous sub-section. However, this is not compulsory since any other alternative (e.g. word embeddings, NNs, LSMTs, etc) can be used.

[3]Generally, we start from the top level of label hierarchy and we proceed by going downwards to the lower levels

[4]Instead of multiplication any other mathematical operation (i.e. summation, division, subtraction) or weighted average/trade-off can be used as an alternative.

Secondly, we suggest another technique that is able to take into account the previous outputs in order to enhance the forthcoming predictions. In contrast with the previous approach, this method is much simpler and straightforward since does not use any extra masking component or mathematical operations. Instead, it merely concatenates to the input of the next layer the output of the prior one. Therefore, by doing so, it discloses the result of the L_{i-1} classifier to the L_i and allows them to participate in the prediction process.

Finally, before the closure, we have to say that can be bind together into a unified method. However, currently, this case is out of our scope and we will not discuss it any further.

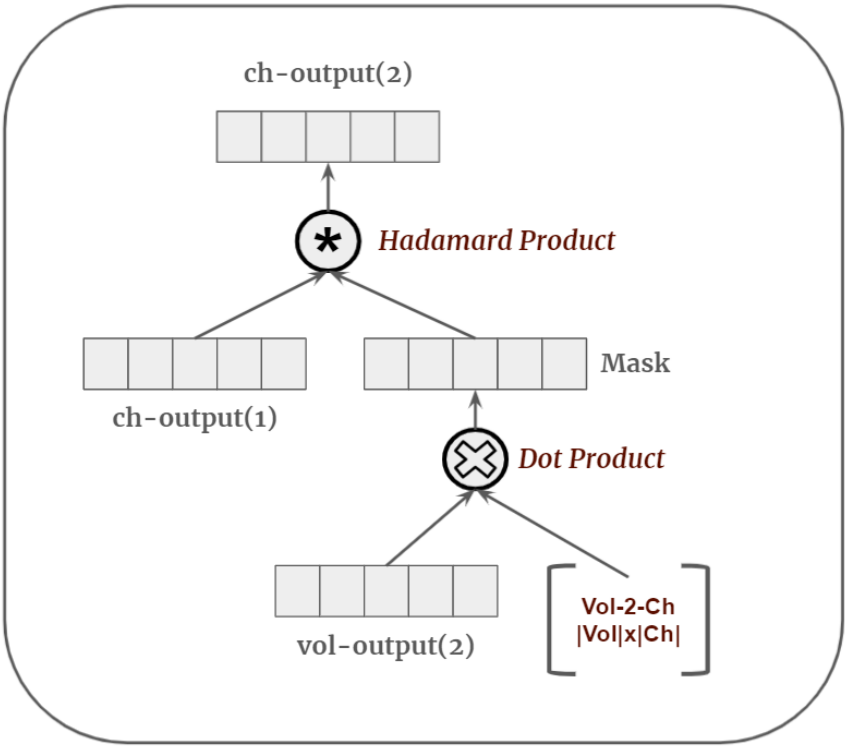


Figure 4.6: The Masking Component for Chapter level.

4.2.3 Other models

Eventually, in this last subsection, we briefly propose some notable suggestions that we detected across the various XMC publications. These ideas are not considered completely autonomous but they can be used in conjunction with the previous models or any other Deep learning technique.

1. **ManyCLS-Tokens:** A very common adjunct that is found in plenty LMTC papers [42, 45] (see Fig- 4.3b). As we said, the CLS token tries to

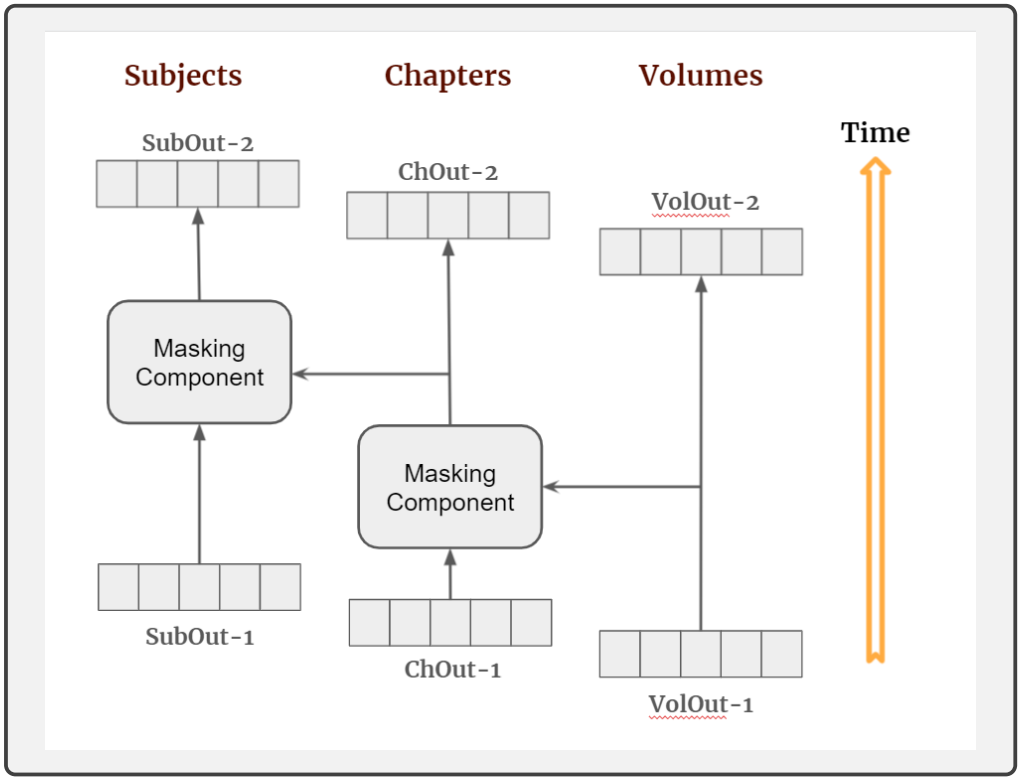


Figure 4.7: Visual representation of the masking process of the Algorithm-1.

Algorithm 1 XMC Hierarchical Masking Algorithm

Require: *The existence of label-hierarchy structure.*
Ensure: *The sequential production of the results.*

Produce:
vol-output # |Batch-Num|∗|Vol-Num| Tensor
ch-output # |Batch-Num|∗|Ch-Num| Tensor
sub-output # |Batch-Num|∗|Sub-Num| Tensor

Tensors:
Vol2Ch # |Vol-Num|∗|Ch-Num|
Ch2sub # |Ch-Num|∗|Sub-Num|

Mask Chapter predictions
Mask-Chapter = vol-output.dot(Vol2Ch) # |Batch-Num|∗|Chp-Num| Tensor
*ch-output = ch-output * Mask-Chapter # Batch-Num|∗|Chp-Num| Tensor*

Mask Subject predictions
Mask-Subject = ch-output.dot(Ch2Sub) # |Batch-Num|∗|Sub-Num| Tensor
sub-output = sub-output ∗ Mask-Subject # |Batch-Num|∗|Sub-Num| Tensor

Return final outputs
Return vol-output, ch-output, sub-output

aggregate the whole textual information of the input into a new feature vector. Nevertheless, when the label set is extremely large then the size of this representation may be poor and insufficient to predict the relevant concepts. A common way to surpass this problem is to use multiple [CLS] tokens. For our experiments, we increase their number to three.

2. **SumLast-BERT:** Similar to SimpleBERT, but instead of the classification token it is used as a feature vector, the average of all the words in the sequence (all except the special [CLS] and [SEP] tokens).
3. **SumLast4-BERT:** Similar to SumLast-BERT, however in this case the average is estimated over the last 4 layers of BERT. [5]

4.3 Evaluation process

4.3.1 MLTC metrics

After the construction of the models, the next step is the selection of appropriate techniques that will be used for the assessment of our algorithms. The evaluation process is an integral component of applied machine learning since it measures the aptitude of the models to perform and generalise on unseen out-of-the-sample data. However, the option of proper metrics is not always obvious especially for the LMTC problem in which the hierarchical structure and the uneven label allocation pose extra difficulties [78] in the procedure. Below we expose the most significant evaluation criteria of LMTC that are relevant to our experiments in the next chapter.

Accuracy: Perhaps the most well-known and intuitive threshold metric. Accuracy is defined as the ratio of the number of correct predictions to the total number of input instances. Complementary to accuracy is the *Error* which measures the rate of incorrect predictions to the total number of input instances

$$Accuracy = \frac{Number of Correct Predictions}{Total Number of Predictions} \tag{4.1}$$

In total, this formula despite its simplicity has severe limitations and is virtually inappropriate when there is a considerable disparity between the number of positive and negative labels. In such cases, accuracy is inadequate and fails to measure impartially the performance of the models.

Precision Recall: Two more resilient and reliable evaluation techniques are these of Precision and Recall. Precision or Positive Predictive Value (PPV)

[5] Instead of the average sum it can be adopted any other method for the production of the feature vector (e.g. concatenation, multiplication, weighted sum, etc).

expresses the proportion of the positive predictions that are actually correct. On the other hand, Recall or True Positive Rate (TPR) quantifies the ratio of the actual positive cases that were identified correctly by the algorithms.

Dismally, precision and recall usually functions conversely. This means that any improvement in precision entails a decrease in recall and vice versa. As a result, in most of cases there is a trade-off to determine which factor of the two is more desirable. At last but not least, we have to mention that Accuracy, Precision and Recall are measures that used exclusively for binary classification problems.

$$Precision = \frac{TP}{TP + FP} \quad (4.2) \qquad Recall = \frac{TP}{TP + FN} \quad (4.3)$$

Where TP, TN, FP and FN comes from Figure 4.8

TP : True Positive, Predicted True and True in reality

TN : True Negative, Predicted False and False in reality

FP : False Positive, Predicted True and False in reality

FN : False Negative, Predicted False and True in reality

:

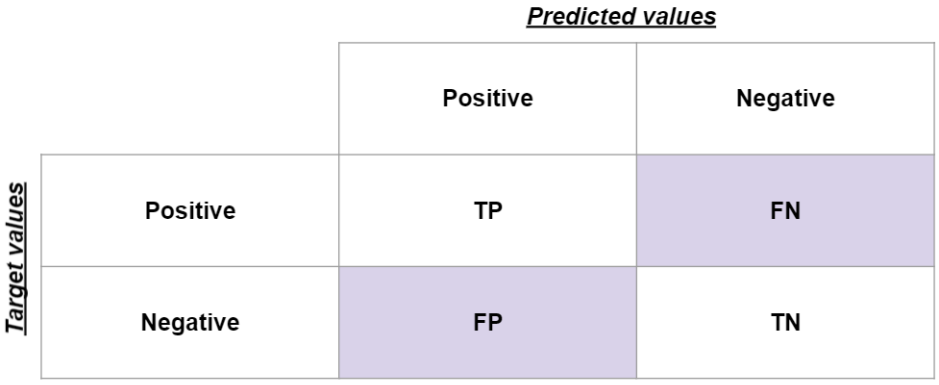


Figure 4.8: A 2x2 Confusion Matrix is a table that describes the performance of an algorithm on a set of data for which the true values are known. Generally, it visualizes the number of the total instances that are correct or incorrect classified.

F1-score is the harmonic mean or the weighted average between precision and recall. Its value ranges in the interval [0, 1], which implies that it reaches its best score at 1 and the worst at 0 respectively.

$$F - Measure = \frac{2 * Precision * Recall}{Precision + Recall} \quad (4.4)$$

Although *Precision, Recall, and F1-Score* are constructed particularly for binary classification cases, they can also be adjusted on the multi-class problem. On the whole, there are three different scenarios to accomplish that:

- **Macro** this measure estimates the harmonic mean for each class independently and then delivers as output the collective arithmetic average. This insinuates that all the classes are enabled to contribute equally to the final result.
- **Micro** aggregates the contributions of all classes to compute the average metric. In other words, it sums up the individual TP, FP, and FN of different classes and then estimates the output. This tough reduces the influence of the rare labels on the overall score.
- **Weighted** similar to the *Macro* metric, it calculates the score for each class independently but in this case, it weights them considering the proportion for each label in the dataset.

Heretofore, we have seen how we can apply the above measurements to binary and multi-class classification problems. However, variants of these methods are regularly used for multi-label classification and especially in LMTC. Precision at top-K predictions (P@K), Recall at top-K predictions (R@K), micro-averaged F1-Score over all labels, and Normalised Discounted Cumulative Gain at top-K prediction (nDCG@K) are only some typical examples.

$$P@k = \frac{1}{T}\sum_{t=1}^{T}\frac{S_t(K)}{K} \tag{4.5}$$

$$R@k = \frac{1}{T}\sum_{t=1}^{T}\frac{S_t(K)}{R_t} \tag{4.6}$$

$$nDCG@k = \frac{1}{T}\sum_{t=1}^{T}\sum_{k=1}^{K}\frac{2^{S_t(K)}}{log(1+k)} \tag{4.7}$$

Where $S_t(K)$ stands for the correct labels amongst top-K

These metric formulas evaluate a corpus of T documents based on the top K-selected labels per document. The reasoning behind them is that since the concept set in XMC task is extremely large, all the instances will be relevant to some (K) labels. So we are interested to know how the algorithm performs in the first (top) predictions. Common values for K are 1, 3, and 5.

It is needless to say that such tactics may undermine or overestimate the real performance of the algorithms since the training instances do not have all the same number of labels.

4.3.2 Raptarchi47k Metrices

At large, there are a plethora of other alternatives [17, 78] that can be used as a substitute to the above methods. Unfortunately, due to the problem's complexity and the large number of different approaches that exist in the bibliography, there is no universal rule for the default selection of the proper metric. Instead, this choice depends on the dataset and the algorithmic structure that is employed to address the task.

Auspiciously, for RAPTARCHIS47k, the standard three-level hierarchy and the fact that there is exactly one and only one active class per layer alleviate significantly the above perplexities. First of all, if we choose a naive flat approach, it is obvious that the K parameter of the Equation 4.5, Equation 4.6, and Equation 4.7 should be equal to 3. This happens because if it is bigger/smaller than 3, then the real performance of the model will be constantly underestimated/overestimated since the metric will consider more/less labels than necessary. Thus, this miscalculation will definitely lead to sub-optimal unrealistic evaluation.

However, as we have explained in Section 4.2, in our methods we try to take advantage of the hierarchical structure of the dataset. So instead of using simple flat approaches, we disassemble the initial problem into three successive multiclass classification sub-tasks. This event allows the utilization of regular TC metrics like *Precision, Recall, and F1-Score* on every level of the hierarchy. This strategy is proven exceptionally beneficial since we can compare directly our outputs with other results from previous studies [2] that confront the problem from the text classification perspective.

Lastly, we expand the simple accuracy of Equation 4.1 so as to expresses the proportion of correct predictions not only at a specific layer but also for all hierarchical levels as a whole. This practically means that a training sample is considered correct only if it has 3 correct predictions (for Volume, Chapter, and Subject categories). We will refer to this metric as ***Cumulative-Accuracy*** or ***CumAcc*** and we will use it to measure the total performance of our techniques over all the layers of the Raptarchis graph. We believe that this approach is extremely useful exceptionally in cases in which we want to know if the model learns to exploit knowledge from the previous layers or the outputs on each level are unrelated and independent.

The Figure 4.9 illustrates all the cases in which a prediction for a unique training sample is considered valid for ***CumAcc*** metric. For Raptarchis47k dataset specifically, this occurs when we have correct outputs for all the *Volume, Chapter, and Subject (3/3)* classes of the hierarchy.

Volume	Chapter	Subject	CumAcc
Correct	**Correct**	**Correct**	**Correct (3/3)**
Correct	Incorrect	Correct	Incorrect (2/3)
Correct	Correct	Incorrect	Incorrect (2/3)
Correct	Incorrect	Incorrect	Incorrect (1/3)
Incorrect	Correct	Correct	Incorrect (2/3)
Incorrect	Incorrect	Correct	Incorrect (1/3)
Incorrect	Correct	Incorrect	Incorrect (1/3)
Incorrect	Incorrect	Incorrect	Incorrect (0/3)

Figure 4.9: *CumAcc* metric counts as correct only the instances which have right predictions for all the levels of the hierarchy.

5. EXPERIMENTS

In this chapter, we describe the implementation and the course of our experiments. We discuss practical programming aspects of this dissertation and provide special details about the code and the design of the algorithms. Then we present our outcomes and compare them with the finding and results of other studies. Ultimately, in the end, we give some general conclusions and observations about the advantages and the drawback of the entire process as a whole.

5.1 Implementation Details

5.1.1 Code's structure

The code of this thesis was written in python 3.7 and deployed and organized in Jupiter notebooks on Google Colaboratory. All the methods in the training stage were implemented with Pytorch [1] while for the BERT-based models were used pre-trained transformer versions from hugging face[2]. Moreover, for the acceleration of our experiments, we heavily relied on the GPU capabilities that are free provided by Google Colab. In general, the overall code of this project can be partitioned into three phases:

1. **Data preparation:** In this step, we create the textual data that will be used for training by concatenated the *title* and *articles* properties of the JSON files in RAPTARCHIS47k dataset. The preprocessing of the raw textual inputs includes only the removal of the special *Line Feed ("$\backslash n$")* characters and of all the redundant white spaces. Also, in order to avoid the unnecessary repetition of this process, we store the data and their categories as *Scikit-learn pickles* [3] so as to retain and preserve them for future use.
2. **Models implementation:** After that, we continue with the construction of our algorithms that are based on the Pytorch and Hugging face libraries. The models contain a Greek or a Multi-lingual *bert-base-uncased* version of

[1] See Pytorch DL framework https://pytorch.org/.
[2] See Hugging face: https://huggingface.co/.
[3] See Scikit-learn pickles.

Google's BERT pre-trained LM component. Then there are 3 simple linear classifiers that map BERTs' embeddings vectors into log probabilities. The total cost is calculated as the cumulative sum of CrossEntropyLosses[4] of these 3 classifiers.

3. **Training/Evaluation:** Finally we employ the TRAIN, DEV, and TEST sets of RAPTARCHIS47k for the training, evaluation, and testing of our techniques. Unfortunately due to the severe GPU limitations[5] of Google, we could neither dedicate more than 3 Epochs to each method nor repeat multiple times our experiments. However, as we will see later in Section 5.2 and despite these restrictions, the models seem to perform satisfactorily as the results are comparable to other studies [2] relevant to the problem.

5.1.2 Models

During the conduction of our experiments, we developed a series of various algorithms established on the ideas and methodologies that are described in Section 4.2. Besides, seeing that these approaches are complementary to each other, we can experiment with them by blended them together in numerous ways. However, it would be rather ineffective to reproduce and evaluate all of them, especially if we take into account the severe limitations and restrictions that are imposed by Google's GPU policy. In opposition, in order to avoid this obstruction we choose to concentrate selectively on the most interesting and alluring combinations according to our opinion. So, we have the following cases:

I ***SimpleBERT, Last3, OneByFour, InTriples, Hybrid:*** The models of this category rely exclusively on the methodology that was described in Subsection 4.2.1. In reality, these approaches are considered the ground floor and the main body of this dissertation.

II ***SumLast, SumLast4, SimpleBERT-3CLS:*** A modification of the former case that incarnates the ideas of Subsection 4.2.3.

III ***MaskSimpleBERT, MaskSumLast4, Mask2SumlLast:*** This cluster comprises the fusion of the two previous categories with the masking techniques that we suggested in Subsection 4.2.2. Generally, the *"masking"* refers to the use of the component of Figure 4.6. The only exception is the *Mask2SumLast* model in which the *"masking"* is simply the usage of the predictions the previous level to the inputs of the next (our second masking technique) [6].

IV ***Mask-Last3-3CLS:*** A superclass model that compounds elements from all the previous cases.

In Table 5.1 is displayed the exact configuration of the above algorithms

[4] The CrossEntropyLoss criterion binds together the LogSoftMax and NLLLoss functions.
[5] See Googles' GPU limitations
[6] See Second-Masking-Technique

Table 5.1: The parameter configuration of the algorithms.

Hyper-parameter	***Setting***
MAX-SEQUENCE-LENGTH	*400*
BATCH SIZE	*4*
MAX PATIENCE	*2*
EPOCHS	*3*
LEARNING RATE	*1e-5*
OPTIMIZER	*Adam*
LOSS FUNCTION	*CrossEntropyLoss*
SCORE METRICS	*Acc, Precision, Recall, F1-Score*
BERT	*'bert-base-multilingual-uncased'* *'nlpaueb/bert-base-greek-uncased-v1'*

All the models share the same configuration

Table 5.2: A brief overview of our experiments.

Experiments Overview			
Category	**Name**	**Dataset**	**BERT-Version**
Baseline	SimpleBERT	FreqNum-[10 , 50]	GreekMulti-lingual
Guided Training	Last3	FreqNum-[10 , 50]	Greek, Multi-lingual
	One-by-Four	FreqNum-[10]	Greek, Multi-lingual
	In-Triples	FreqNum-[10]	Greek, Multi-lingual
	Hybrid	FreqNum-[10]	Greek, Multi-lingual
Other	SumLast	FreqNum-[10 , 50]	Greek
	SumLast4	FreqNum-[50]	Greek
	SimpleBERT-3CLS	FreqNum-[10]	Greek
Masking	Mask-SimpleBERT	FreqNum-[10 , 50]	Greek, Multi-lingual
	Mask-SumLast4	FreqNum-[10]	Greek
	Mask2-SumLast	FreqNum-[10]	Greek
	Mask-SumLast4	FreqNum-[10]	Greek
All-Cases-Combination	Mask-Last3-3CLS	FreqNum-[10]	Greek

The 4 categories of our algorithms with their predefined hyper-parameters

5.2 Results

After the construction of the models, we proceed with the execution of the experiments. As we mentioned before, we utilize the *FreqNum-10* and *FreqNum-50* datasets for training, and then we evaluate the outputs of the model by using the evaluation metrics described in Section 4.3.

The Table 5.2 showcases the exact outline of our experiments. In a nutshell, the *"Category"* column of the table indicates the class in which each technique belongs to, the *"Name"* is the programming name that we choose for the model, and the *"Dataset"* declares the datasets on which the models were tested. Eventually, the BERT-Version" implies the BERT component, Monolingual (Greek) or Multilingual, that was used for the text representation of the textual input.

Lastly, on the tables Table 5.3 and Table 5.4 there is a concise congregation of the final results that emerged from the conduction of the experiments on *FreqNum-10* and *FreqNum-50* datasets.

5.2.1 FreqNum-10 dataset

From Table 5.3 we notice that for *FreqNum-10* dataset the models that use the Greek monolingual version of BERT in almost all cases outperform their Multilingual competitors. This fact becomes more noticeable if we compare the outputs that deliver the two BERT variations for the *"Guided Training"* methods. Indeed, we can see that all the greek models of this category are able to offer substantially better results than their cross-lingual counterparts. This fact verifies our intuition that Transformer-based models like *GREEK-BERT* that are pre-trained exclusively on vast greek corpora are more suitable for training and fine-tuning on datasets like RAPTARCHIS47k.

Going forward, we can observe that our masking techniques can further improve and corroborate the overall training ability of the algorithms (e.g. see *SimpleBert/MaskSimpleBert* on both greek and multilingual versions and *SumLast-Mask2SumLast*). On these occasions, we witness that although there is a small decline in Accuracy and F1-Score for *Volume* and *Chapter* (the two upper layers of the hierarchy) labels, the Acc and F1-Score for *Subject* label is increased significantly. This means that the models become more coherent in their predictions and as a result, we have essential improvement for *Cumulative Accuracy* and thus on the total performance as a whole.

At last but not least, it is apparent that the utilization of extra [CLS] tokens (e.g. *SimpleBert-3CLS*) may enhance the learning capabilities of the models and in this case to produce the highest outcome for CumAcc and F1-score. We will examine and analyze this finding later in this chapter.

5.2.2 FreqNum-50 dataset

Proceeding on Table 5.4 we behold the outputs for *FreqNum-50* dataset. This case is much easier than the previous one since the size of the data collection and of the label tree/set is smaller than the *FreqNum-10* scenario.

Like above, our first observation is that monolingual models are more robust than the cross-lingual versions. However here we see that masking deteriorates the performance for the Greek version while it improves it for Multilingual. This may happen because the outputs of the Greek edition are already high enough (Acc: 0.83 [0.93, 0.89, 0.86]). Masking on the other hand, like earlier, reduces the Acc for the Volume layers and hence it harms the overall performance. On the contrary, in the multilingual scenario masking hurts slightly the Volume-Accuracy whereas it increases meaningfully the Subject-Accuracy and therefore the total *CumAcc*. Altogether, we can say that masking is unnecessary for cases in which the performance is extremely high while it is more beneficial for cases like the second one where the results remain low and dissatisfied.

For the rest of the Greek methods, we note that their results are comparable with the *SumLast* model to achieve the best performance and the *SumLast4* the lowest.

Eventually, at this point, we have to stress again that the difference in the performance between the models in *FreqNum-10* and *FreqNum-50* is own to the difference in their size (see Table 3.1). Indeed the task is much easier for the latter case where there are fewer input data and a smaller label-set available for training instead of the former case in which the dataset is increased rapidly and the label-set approximates the standards of the other XMTC datasets.

Table 5.3: The table with the aggregated results for the FreqNm:10 case.

Models	Loss	Acc	Macro-F1
Greek-BERT			
SimpleBERT	2.79 [0.43, 0.80, 1.55]	0.65 [0.90, 0.83, 0.67]	[0.88, 0.74, 0.
Last3	2.60 [0.43, 0.79, 1.38]	0.67 [0.89, 0.82, 0.70]	[0.87, 0.74, 0.
One-By-Four	2.43 [0.46, 0.79, 1.16]	0.68 [0.87, 0.79, 0.69]	[0.84, 0.70, 0.
In-Triples	2.25 [0.45, 0.74, 1.06]	0.69 [0.88, 0.82, 0.74]	[0.85, 0.71, 0.
Hybrid	2.46 [0.46, 0.81, 1.18]	0.67 [0.88, 0.81 , 0.72]	[0.86, 0.73, 0.
Mask-SimpleBERT	3.38 [0.59, 0.99, 1.79]	0.67 [0.87, 0.81, 0.72]	[0.83, 0.73, 0.
SimpleBERT-3CLS	2.04 [0.40, 0.63, 0.99]	0.74 [0.90, 0.85, 0.77]	[0.89, 0.81, 0.
Mask-SimpleBERT-3CLS	3.24 [0.60, 0.89, 1.74]	0.68 [0.86, 0.82, 0.75]	[0.81, 0.77, 0.
Mask-Last3-3CLS	4.60 [0.54, 1.98, 2.07]	0.61 [0.86, 0.75, 0.70]	[0.83, 0.61, 0.
SumLast	2.79 [0.45, 0.81, 1.52]	0.64 [0.89, 0.82, 0.67]	[0.88, 0.74, 0.
Mask2-SumLast	2.75 [0.45, 0.82, 1.47]	0.66 [0.90, 0.83, 0.69]	[0.88, 0.75, 0.
Mask-SumLast4	2.99 [0.51, 1.23, 1.24]	0.69 [0.88, 0.81, 0.72]	[0.85, 0.77, 0.
MultiLingual-BERT			
SimpleBERT	6,45 [1.02, 2.05, 3.38]	0,29 [0.74, 0.55, 0.34]	[0.67, 0.30, 0.
Last3	3.24 [0.51, 0.95, 1.77]	0.58 [0.87, 0.77, 0.62]	[0.85, 0.66, 0.
One-By-Four	3.40 [0.66, 1.03, 1.71]	0.58 [0.83, 0.76, 0.66]	[0.78, 0.63, 0.
In-Triples	2.96 [0.58, 0.98, 1.39]	0.60 [0.84, 0.76, 0.67]	[0.80, 0.64, 0.
Hybrid	3.06 [0.56, 1.02, 1.46]	0.61 [0.85, 0.75, 0.66]	[0.82, 0.65, 0.
Mask-SimpleBERT	3.74 [0.98, 1.21, 1.53]	0.62 [0.78, 0.73, 0.69]	[0.65, 0.61, 0.

RAPTARCHIS47k–FreqNum

Table 5.4: The table with the aggregated results for the FreqNm:50 case.

Models	Loss	Acc	Macro-F1
Greek-BERT			
SimpleBERT	1.30 [0.30, 1.30, 0.57]	0.83 [0.93, 0.89, 0.86]	
Last3	3.29 [0.29, 0.45, 0.54]	0.83 [0.93, 0.89, 0.85]	
Mask-SimpleBERT	1.98 [0.61, 0.74, 0.63]	0.77 [0.86, 0.89, 0.85]	
Sum-Last	1.33 [0.30, 0.47, 0.56]	0.84 [0.93, 0.90, 0.86]	[0.91, 0.84, 0.86]
Sum-Last4	1.56 [0.35, 0.55, 0.65]	0.81 [0.92, 0.87, 0.83]	[0.87, 0.81, 0.85]
MultiLingual-BERT			
SimpleBERT	3.62 [0.79, 1.30, 1.52]	0.58 [0.81, 0.71, 0.63]	[0.75, 0.57, 0.62]
Mask-SimpleBERT	3.74 [0.98, 1.21, 1.53]	0.62 [0.78, 0.73, 0.69]	[0.65, 0.61, 0.67]

Raptarchis's-Dataset–FreqNum:50

5.3 Analysis and Conclusions

5.3.1 [CLS] Representations

After the termination of the learning procedure and the tuning of the parameters, a lot of questions arise about the inner workings of the BERT component and the differences amongst the various [CLS] representations. In the work of Manginas et al. [2020] , the authors are trying to demystify some of these questions. Thus in order to do so and to increase the explainability of their models, they compare the average angular distances (L_2 normalized norm) between the [CLS] tokens of each one of the 12 BERT layers in the development set of [17].

Along the same line, we adopt a similar approach to deal with these issues and shed light on some aspects of the problem. The only differentiation here is that, instead of the angular distance, we use the cosine similarity to measure the affinity between the [CLS] vectors. Additionally, we expand this strategy for the case of *SimpleBert-3CLS, MaskLast3-3CLS* and generally for all **.-3CLS* models that use multiple classification tokens to solve the downstream MLTC task. Having said that, we can continue with the demonstration of the results that emerged as a byproduct of this procedure.

First of all, the Figure 5.1 appears the exact outcomes that stem from the paper of Manginas et al. [2020] . More specifically, one of their primary conclusions there was that the *Guided training* yielded larger angular distances which subsequently signify better parameter utilization for the BERT models. In our work, we can reproduce and verify the same findings for RAPTARCHIS47k dataset on both the Greek and Multilingual versions of our models.

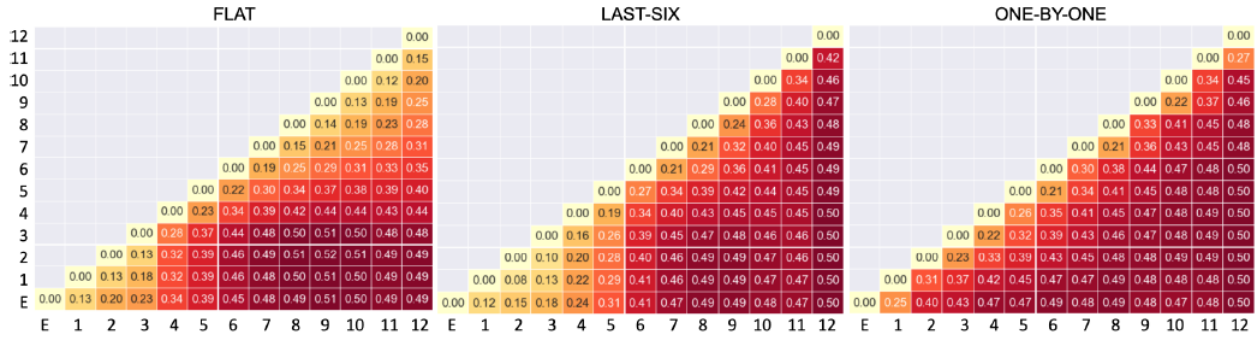


Figure 5.1: Flat–Last6–OneByOne: The [CLS] angular distance as it is presented on the paper of Manginas et al.[5].

Indeed, as we can easily observe on Figure 5.2 and Figure 5.3 our models on Raptarchis have almost identical behavior with these of Figure 5.1. To put it this more simply, the cosine similarity of the [CLS] representations across the 12 layers of BERT declines steadily and thus the classification tokens become

more dissimilar. Actually, this phenomenon in our case is much more intense and discernible due to the shallow hierarchy and the plainness of the dataset (3 instead of 6 levels).

For the multi-lingual models on Figure 5.3 the situation is a little more perplexing. For this case, there is a spread in the cosine dissimilarity across some layers of BERT, however, this does not happen evenly but rather irregularly. So not a safe conclusion can be deducted since it is required more work and research on the area. More details about this topic can be found in the Appendix.

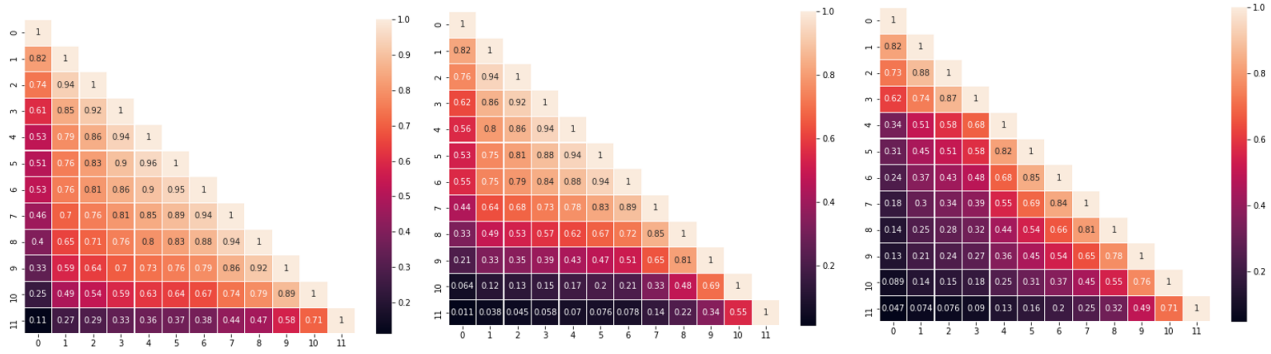


Figure 5.2: Greek: Simple–Last3–OneByFour: The [CLS] cosine similarity for the corresponding to the Figure 5.1 Greek models on Raptarchis47k.

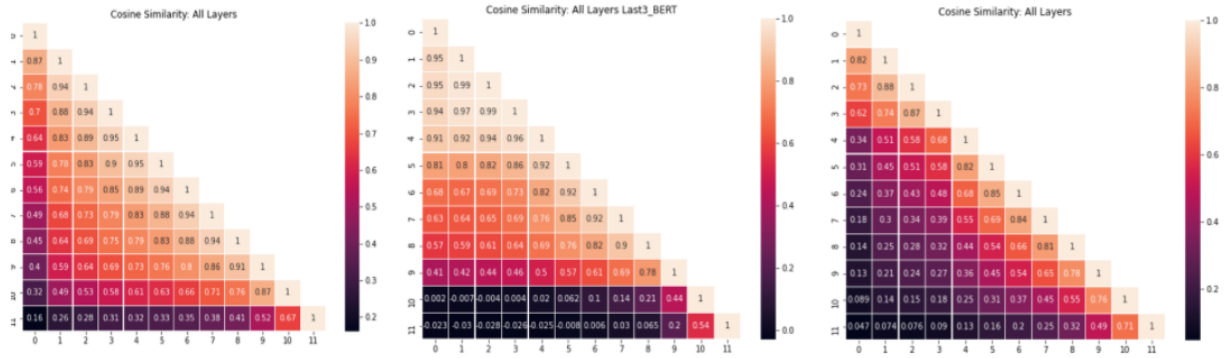


Figure 5.3: Multilingual: Simple–Last3–OneByFour: The [CLS] cosine similarity for the corresponding to the Figure 5.1 multi-lingual models on Raptarchis47k.

5.3.2 Multiple [CLS] tokens

As we have noticed in other works [42, 45] the use of multiple classification tokens at the beginning of the input sequence is an ordinary and efficient method. This technique is usually applied on occasions when the label space is too large and the existence of a unique [CLS] vector may not be informative enough to predict correctly all the relevant concepts. We adopt this practice in our experiments so as to reinforce the learning capabilities of our models. Undoubtedly, from Table 5.3 it is conspicuous that this tactic not only can enhance the performance of some methods but also in the case of (SimpleBERT-3CLS provides the best

results against all its other competitors.

An interesting observation here is that despite the fact that the multiple-CLS tokens and the masking techniques improve the overall capabilities of the models, if we bind them together in one model (e.g. *Mask-SimpleBERT-3CLS and Mask-Last3-3CLS*) then this new unified edition performs slightly worse. To clarify this finding we compare the cosine similarity of the three [CLS] representations for *SimpleBERT-3CLS* and *Mask-SimpleBERT-3CLS* methods.

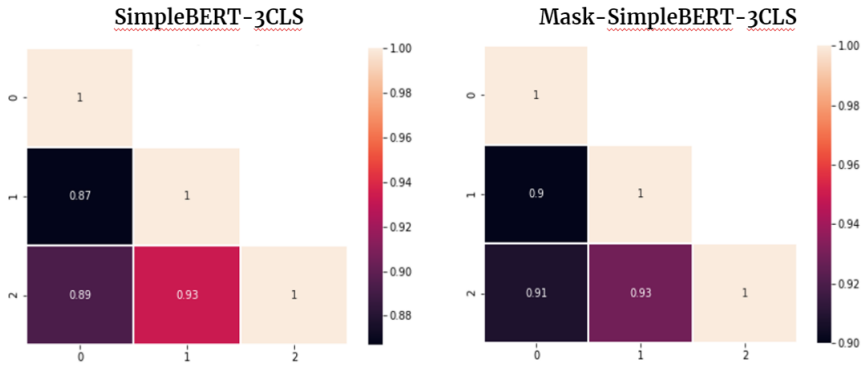


Figure 5.4: The cosine similarity amongst the three classification tokens of the *SimpleBERT-3CLS* and *Mask-SimpleBERT-3CLS* models.

The Figure 5.4 illustrates this comparison. As we can see, the classification tokens for the *SimpleBERT-3CLS* model are lightly more dissimilar, and therefore they can capture and aggregate more information. In opposition, the masking component for *Mask-SimpleBERT-3CLS* seems to impede to [CLS] representations to differentiate from each other and hence hamper their ability to extend their expressive power.

5.3.3 The *Mask-SimpleBERT-3CLS* case

Similarly to the above comparison, we proceed by exploring the [CLS] vectors for the *Mask-SimpleBERT-3CLS* model. The image Figure 5.5 shows the cosine similarities between the classifications tokens on the last three layers (with the layer [-3] to correspond to Volumes, the [-2] to Chapters, and the [-1] to Subjects). We observe that for the higher levels, on which we get the finest results, the vectors are more dissimilar while as we go down to the lower level, on which we get lower results, the vectors become more identical. This means that the relation between vector similarity and learning competence is inversely proportional for this model.

5.4 Further discussion

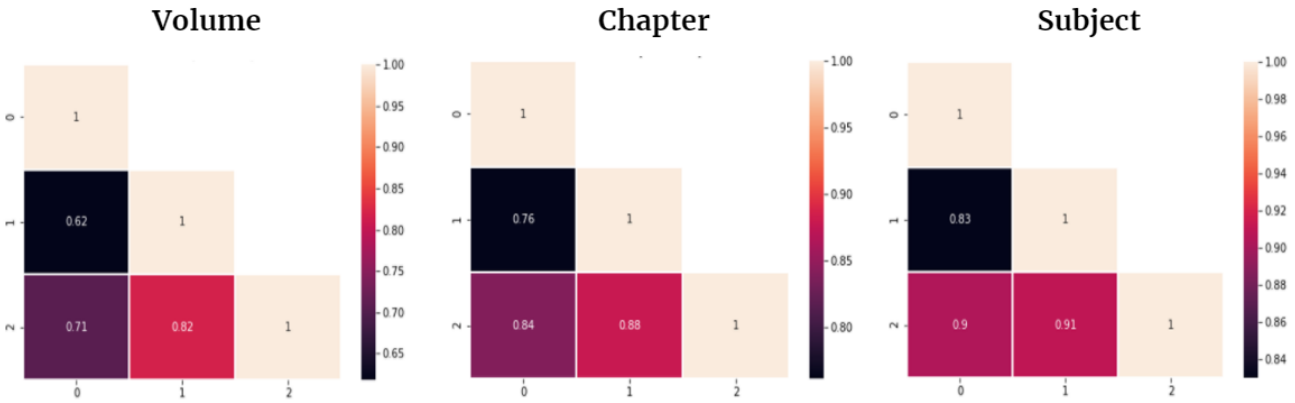


Figure 5.5: Greek: Mask-Last3-3CLS: The cosine similarity amongst the 3 [CLS] vectors for the last 3 layers of the BERT component.

5.4.1 Comparison with Papaloukas's Results

After the presentation and the analysis of our results, the next question that arises is how effectual are our methods in comparison with other solutions affiliated to the topic. Unfortunately, the only work that we can use for this purpose is this of Papaloukas [2020] who confronts from the aspect of the multi-class Text Classification aspect. In this slightly different version of the problem, the author implements various algorithms which are solely responsible for a distinct concept of the category set.

For reasons that have already been explained in Chapter 4 the comparison between the two works is not only feasible but also reasonable and desirable. Moreover, since we know from the previous subsections that our methods perform satisfactorily on the first level of the label tree, we will turn our attention to the next two layers of hierarchy. So for the next few paragraphs we will focus exclusively on the *Chapter* and *Subject* levels.

For the convenience and conciseness of this process, we mention the outputs of some of our algorithms and we adjust the evaluation metrics appropriately so as to be able to be compared with the results of Papaloukas.

	CHAPTER								
	ALL LABELS			FREQUENT			FEW-SHOT		
	P	R	F1	P	R	F1	P	R	F1
SVM-BOW	77.9	77.9	77.9	77.9	78.6	78.2	90.0	09.3	16.8
XGBOOST-BOW	67.5	67.5	67.5	67.8	68.1	67.9	19.2	10.3	13.4
BIGRU-MAX	77.5	77.5	77.5	77.9	77.9	77.9	44.9	45.4	45.1
BIGRU-ATT	81.1	81.1	81.1	81.1	81.6	81.3	86.7	40.2	54.9
BIGRU-LWAN	76.8	76.8	76.8	76.9	77.3	77.1	63.8	30.9	41.7
BERT-BASE-ML	82.4	82.4	**82.4**	82.4	82.7	**82.6**	84.1	54.6	**66.3**
XLM-ROBERTA	81.0	81.0	81.0	81.0	81.4	81.2	78.7	38.1	51.4
GREEK-BERT	81.4	81.4	81.4	81.4	81.8	81.6	81.3	40.2	53.8

Figure 5.6: The exact results for *Chapter* category according to Papaloukas [2020] .

Table 5.5: Our results for *Chapter* category for *Micro-P/R/F1, Macro-F1 and Weighted-F1*.

Models	Micro-P	Micro-R	Micro-F1	Macro-F1	Weighted-F1
Chapters					
SimpleBERT	0.83	0.83	0.83	0.75	0.82
Hybrid	0.81	0.81	0.81	0.73	0.81
One-By-Four	0.80	0.80	0.80	0.70	0.79
Last-3	0.82	0.82	0.82	0.74	0.82
SumLast	0.82	0.82	0.82	74	0.82
Mask-SimpleBERT	-	-	-	78	0.83

RAPTARCHIS47k–FreqNum:10

Table 5.6: Our results for Subject category for *Micro-P/R/F1, Macro-F1 and Weighted-F1*.

Models	Micro-P	Micro-R	Micro-F1	Macro-F1	Weighted-F1
Subjects					
SimpleBERT	0.68	0.68	0.68	0.52	0.64
Hybrid	0.72	0.72	0.72	0.64	0.70
One-By-Four	0.69	0.69	0.69	0.57	0.66
Last-3	0.71	0.71	0.71	0.58	0.67
SumLast	0.68	0.68	0.68	0.52	0.64
Mask-SimpleBERT	0.74	0.74	0.74	68	0.73

RAPTARCHIS47k FreqNum:10

	SUBJECT								
	ALL LABELS			FREQUENT			FEW-SHOT		
	P	R	F1	P	R	F1	P	R	F1
SVM-BOW	37.9	37.9	37.9	37.9	47.8	42.3	00.0	00.0	00.0
XGBOOST-BOW	55.3	55.3	55.3	56.1	64.8	60.1	46.9	19.1	27.2
BIGRU-MAX	62.9	62.9	62.9	66.0	70.5	68.1	47.1	37.8	42.0
BIGRU-ATT	74.8	74.8	74.8	75.3	79.6	77.4	72.6	61.1	66.3
BIGRU-LWAN	65.2	65.2	65.2	68.1	72.8	70.4	50.7	40.4	45.0
BERT-BASE-ML	79.5	79.5	**79.5**	81.6	84.2	**82.9**	70.9	66.5	68.6
XLM-ROBERTA	63.5	63.5	63.5	69.3	70.8	70.1	40.1	39.1	39.6
GREEK-BERT	79.3	79.3	79.3	80.8	83.4	82.1	73.3	68.7	**70.9**

Figure 5.7: The exact results for *Subject* category from Papaloukas [2020] .

So, the Figure 5.6 and Figure 5.7 disclose the exact results as they are presented and analyzed by Papaloukas [2020] . In the figures, it is measured the performance of different AI models on *Chapter* and *Subject* categories for the the *Micro-{ Precision, Recall and F1-Score }*. The process is materialized for the *All, Frequent and Few-shot* scenario. More details about this procedure can be found in [2].

At this stage, we use the *Micro-{ Precision, Recall and F1-Score }* as well as the *Macro/Weighted-F1-Score* measurements in order to establish a stable and correct matching between the two works. Our corresponding findings can be seen in the Table 5.5 and the Table 5.6.

Overall, from the comparison, it is clear that although we neither utilize the same amount of parameters nor employ any category-specific technique, our methods seem to work satisfactorily and to be close to their competitors. Actually, we observe that with only a few epochs the produced performance is sufficient enough for *Chapters* and a little lower for *Subjects* level. This gap in the results however is physical and expectable since label-specific models are more straightforward and have more training advantages over their XMTC counterparts.

5.4.2 Advantages/Disadvantages and Synopsis

All in all, and as we have seen from the previous paragraphs, despite the fact that the training of the models lasts only for a few iterations, this is enough for some layers of the label tree to achieve competitive and sound outcomes. However, the small number of the epochs impedes the convergence and does not allow us to draw safe and absolute conclusions about the real capabilities and potentials of our techniques.

Furthermore, a major drawback in this work comes from the nature of our dataset. This happens mainly due to the fact that although Raphtarchis 47k is an organized data collection with a very convenient and specific structure, its innate hierarchy hurdle the generalization to other more recent and intricate legislation which encompasses many unrelated topics clustered altogether under the same amendment or law (e.g. omnibus bills, etc.)

Finally, for the real assessment and evaluation of our methods, it is necessary their test on more challenged and complicated XMTC datasets. Datasets will include more complex and larger label graphs as well as a greater number of available text documents and corpora.

6. SUMMARY AND FUTURE WORK

In this thesis, we engaged with the problem of extreme multi-label classification for greek legal documents. We commenced from the traditional approaches and went through to the most modern and state-of-the-art works that have been published the past few years. We explored RAPTARCHIS47k dataset, a novel dataset suitable for research on legal TC and MLTC problems, and we excavate and reveal new findings related to its inner hierarchy and nature. Finally, we expand the current bibliography on XMTC by proposing new techniques and testing other innovative ideas on our dataset.

In the future, there are several directions in light of our study. In the first place, it would be more than desirable to exist a constant update and refinement of the Raptarchis47k dataset so as to include all the available Greek legal code and other special cases of legislation like omnibus bills (cases that do not be covered on the current version of the dataset). Beyond that, we intend to continue the investigation of our methods and to probe their performance and behavior on other more challenging data collections.

All things considered, we expect that the current study will boost the Greek NLP research in the XMTC field while at the same time it would provide a new strong baseline for future study and experimentation in the area.

7. ABBREVIATIONS-ACRONYMS

AI	Artificial Intelligence
BERT	Bidirectional Encoder Representations from Transformers
BGRU	Bidirectional Gated Recurrent Unit
BLSTM	Bidirectional Long Short Term Memory
BOW	Bag Of Words
DAG	Directed Acyclic Graph
GLC	Greek Legislation Code
GRU	Gated Recurrent Unit
JSON	JavaScript Object Notation
LM	Language Model
LMTC	Large Multi-label Text Classification
LSTM	Long Short Term Memory
ML	Machine Learning
MLM	Masked Language Model
MLTC	Multi-label Text Classification
NER	Name Entity Recognition
NLI	Natural Language Inference
NLP	Natural Language Processing
NN	Neural Network
NSP	Next Sentence Prediction
PoS	Part Of Speech Tagging
PPV	Positive Predicted Value
QA	Question Answering

RNN	Recurrent Neural Network
SOTA	State Of The ART
TC	Text Classification
TF-IDF	Term Frequency-Inverse Term Frequency
TPR	True Positive Rate
XMC	Extreme Multi-label text Classification
XMTC	Extreme Multi-label Text Classification

A. APPENDIX

A.1 Raptarchis Dataset

A.1.1 Raptarchis DAG

As we have already explained in the corresponding subsection, the thematic structure of Raptarchis47k is more intriguing than it may be expected. This practically means that in order to shed light on its deeper structure hierarchy we have to know its internal relations and interconnections among the classes of the label set. In the next following paragraphs, we will exhibit the most fundamental associations of the concepts on the label graph.

For the simplicity of this process, we break down our findings into three parts. The first one includes all the *Chapter* labels which have multiple *Volume* parents and, similarly, the second has all the *Subject* with many *Chapter* parents.

A more detailed and meticulous presentation of these results can be found on the Github of the thesis can be found here

————- Chapters-With-Multiple-Parents ————-

1. ΔΙΑΦΟΡΑ: len=2, [’ΓΕΩΡΓΙΚΗ ΝΟΜΟΘΕΣΙΑ’, ’ΕΚΚΛΗΣΙΑΣΤΙΚΗ ΝΟΜΟΘΕΣΙΑ’]
2. ΔΙΑΦΟΡΑ ΑΣΦΑΛΙΣΤΙΚΑ ΤΑΜΕΙΑ: len=2, [’ΑΣΦΑΛΙΣΤΙΚΑ ΤΑΜΕΙΑ’, ’ΝΟΜΙΚΑ ΠΡΟΣΩΠΑ ΔΗΜΟΣΙΟΥ ΔΙΚΑΙΟΥ’]
3. ΕΙΔΙΚΕΣ ΔΙΑΔΙΚΑΣΙΕΣ: len=2, [’ΠΟΙΝΙΚΗΝΟΜΟΘΕΣΙΑ’, ’ΠΟΛΙΤΙΚΗ ΔΙΚΟΝΟΜΙΑ’]

Total Chapters With Multiple Parents: 3

———————- Subjects-With-Multiple-Parents ———————-

1. ΟΡΓΑΝΙΣΜΟΣ: len=2, [’ΥΠΟΥΡΓΕΙΟ ΕΣΩΤΕΡΙΚΩΝ ΔΗΜ.ΔΙΟΙΚΗΣΗΣ ΚΑΙ ΑΠΟΚΕΝΤΡΩΣΗΣ’, ’ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ’]
2. ΒΟΣΚΟΤΟΠΟΙ: len=2, [’ΑΓΡΟΤΙΚΕΣ ΜΙΣΘΩΣΕΙΣ’, ’ΝΟΜΙΚΑ ΠΡΟΣΩΠΑ ΚΑΙ ΕΚΜΕΤΑΛΛΕΥΣΕΙΣ’]
3. ΟΡΓΑΝΙΚΕΣ ΔΙΑΤΑΞΕΙΣ: len=11, [’ΠΑΝΕΠΙΣΤΗΜΙΟ ΙΩΑΝΝΙΝΩΝ’, ’ΥΠΟΥΡΓΕΙΟ ΔΗΜΟΣΙΑΣ ΤΑΞΗΣ’, ’ΟΡΓΑΝΙΣΜΟΣ ΠΟΛΕΜΙΚΗΣ ΑΕΡΟΠΟΡΙΑΣ’, ’ΑΝΩΤΑΤΗ ΣΧΟΛΗ ΚΑΛΩΝ ΤΕΧΝΩΝ’, ’ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΟΝΙΚΗΣ’, ’ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΑΤΡΩΝ’, ’ΥΠΟΥΡΓΕΙΟ ΕΘΝΙΚΗΣ ΑΜΥΝΑΣ’, ’ΠΥΡΟΣΒΕΣΤΙΚΟ ΣΩΜΑ’, ’ΥΠΟΥΡΓΕΙΟ ΠΟΛΙΤΙΣΜΟΥ’, ’ΤΕΛΩΝΕΙΑΚΗ ΥΠΗΡΕΣΙΑ’, ’ΥΠΟΥΡΓΕΊΟ ΚΟΙΝΩΝΙΚΏΝ ΑΣΦΑΛΊΣΕΩΝ’]
4. ΓΕΝΙΚΕΣ ΔΙΑΤΑΞΕΙΣ: len=35, [’ΠΡΟΣΤΑΣΙΑ ΚΑΙΚΙΝΗΤΡΑ ΙΔΙΩΤΙΚΩΝ ΕΠΕΝΔΥΣΕΩΝ’, ’ΣΤΡΑΤΟΛΟΓΙΑ’, ’ΠΡΟΣΩΡΙΝΕΣ ΑΤΕΛΕΙΕΣ’, ’ΠΡΟΣΩΠΙΚΟ ΤΑΧΥΔΡΟΜΕΙΩΝ’, ’ΠΕΡΙΦΕΡΕΙΕΣ’, ’ΙΑΜΑΤΙΚΕΣ ΠΗΓΕΣ’, ’ΟΡΓΑΝΩΣΗ ΧΡΟΝΟΥ ΕΡΓΑΣΙΑΣ’, ’ΕΚΤΕΛΕΣ ΔΗΜΟΣΙΩΝ ΕΡΓΩΝ’, ’ΣΧΟΛΕΣ ΠΑΝΕΠΙΣΤΗΜΙΟΥ ΑΘΗΝΩΝ’, ’ΘΗΡΑ’, ’ΔΗΜΟΣΙΟ ΧΡΕΟΣ’, ’ΤΕΛΩΝΕΙΑΚΟΣ ΚΩΔΙΚΑΣ’, ’ΓΕΩΡΓΙΚΟΙ ΣΥΝΕΤΑΙΡΙΣΜΟΙ ΑΓΡΟΤΙΚΕΣ ΣΥΝΕΤΑΙΡΙΣΤΙΚΕΣ ΟΡΓΑΝΩΣΕΙΣ’, ’ΔΗΜΟΣΙΟ ΛΟΓΙΣΤΙΚΟ’, ’ΟΡΓΑΝΙΣΜΟΊ ΚΟΙΝΩΝΙΚΉΣ ΑΣΦΑΛΊΣΕΩΣ’, ’ΔΙΑΜΕΤΑΚΟΜΙΣΗ’, ’ΔΙΟΙΚΗΣΗ ΕΚΠΑΙΔΕΥΣΕΩΣ’, ’ΠΡΟΣΤΑΣΙΑ ΝΟΜΙΣΜΑΤΟΣ’, ’ΣΥΝΕΤΑΙΡΙΣΜΟΙ’, ’ΤΕΛΩΝΕΙΑΚΕΣ ΑΤΕΛΕΙΕΣ’, ’ΤΑΧΥΔΡΟΜΙΚΑ ΤΑΜΙΕΥΤΗΡΙΑ’, ’ΕΓΓΕΙΟΒΕΛΤΙΩΤΙΚΑ ΕΡΓΑ’, ’ΣΩΜΑΤΕΙΑ’, ’ΣΤΕΓΑΣΗ ΔΗΜΟΣΙΩΝ ΥΠΗΡΕΣΙΩΝ’, ’ΥΔΡΕΥΣΗ’, ’ΑΠΟΤΑΜΙΕΥΣΗ’, ’ΑΛΥΚΕΣ’, ’ΠΛΟΗΓΙΚΗ ΥΠΗΡΕΣΙΑ’, ’ΑΝΑΠΗΡΟΙ ΚΑΙ ΘΥΜΑΤΑ ΠΟΛΕΜΟΥ’, ’ΔΗΜΟΣΙΑ ΔΙΟΙΚΗΣΗ’, ’ΕΤΑΙΡΕΙΕΣ ΠΕΡΙΩΡΙΣΜΕΝΗΣ ΕΥΘΥΝΗΣ’, ’ΦΟΡΤΟΕΚΦΟΡΤΩΣΕΙΣ’, ’ΕΡΓΑ ΚΑΙ ΠΡΟΜΗΘΕΙΕΣ ΔΗΜΩΝ ΚΑΙ ΚΟΙΝΟΤΗΤΩΝ’, ’ΕΚΤΕΛΩΝΙΣΤΕΣ’, ’ΛΟΥΤΡΟΠΟΛΕΙΣ’]
5. ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ: len=2, [’ΣΧΟΛΕΣ ΠΑΝΕΠΙΣΤΗΜΙΟΥ ΑΘΗΝΩΝ’, ’ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΟΝΙΚΗΣ’]
6. ΣΧΟΛΗ ΕΠΙΣΤΗΜΩΝ ΥΓΕΙΑΣ: len=2, [’ΣΧΟΛΕΣ ΠΑΝΕΠΙΣΤΗΜΙΟΥ ΑΘΗΝΩΝ’, ’ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΟΝΙΚΗΣ’]
7. ΔΙΕΘΝΕΙΣ ΣΥΜΒΑΣΕΙΣ: len=3, [’ΡΥΘΜΙΣΗ ΤΗΣ ΑΓΟΡΑΣ ΕΡΓΑΣΙΑΣ’, ’ΔΙΚΑΙΟ ΤΩΝ ΠΡΟΣΩΠΩΝ’, ’ΕΙΔΙΚΑΙ ΚΑΤΗΓΟΡΙΑΙ ΠΛΟΙΩΝ’]
8. ΦΙΛΟΣΟΦΙΚΗ ΣΧΟΛΗ: len=3, [’ΠΑΝΕΠΙΣΤΗΜΙΟ ΙΩΑΝΝΙΝΩΝ’, ’ΣΧΟΛΕΣ ΠΑΝΕΠΙΣΤΗΜΙΟΥ ΑΘΗΝΩΝ’, ’ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΟΝΙΚΗΣ’]
9. ΑΘΕΜΙΤΟΣ ΑΝΤΑΓΩΝΙΣΜΟΣ: len=2, [’ΠΡΟΣΤΑΣΙΑ ΤΩΝ ΣΥΝΑΛΛΑΓΩΝ’, ’ΕΚΜΕΤΑΛΛΕΥΣΗ ΘΑΛΑΣΣΙΩΝ ΣΥΓΚΟΙΝΩΝΙΩΝ’]
10. ΑΤΟΜΙΚΑ ΕΓΓΡΑΦΑ ΑΞΙΩΜΑΤΙΚΩΝ: len=2, [’ΣΤΡΑΤΙΩΤΙΚΟΙ ΓΕΝΙΚΑ’, ’ΑΞΙΩΜΑΤΙΚΟΙ ΕΝΟΠΛΩΝ ΔΥΝΑΜΕΩΝ’]
11. ΠΕΡΙΦΕΡΕΙΑΚΕΣ ΥΠΗΡΕΣΙΕΣ[300]: len=2, [’ΥΠΟΥΡΓΕΙΟ ΓΕΩΡΓΙΑΣ’, ’ΔΙΟΙΚΗΣΗ ΚΟΙΝΩΝΙΚΗΣ ΠΡΟΝΟΙΑΣ’] ([69, 149])
12. ΥΠΗΡΕΣΙΑΚΑ ΣΥΜΒΟΥΛΙΑ: len=3, [’ΥΠΟΥΡΓΕΙΟΕΘΝΙΚΗΣ

ΟΙΚΟΝΟΜΙΑΣ', 'ΣΧΟΛΕΣ ΠΑΝΕΠΙΣΤΗΜΙΟΥ ΑΘΗΝΩΝ','ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ']

13. ΤΑΜΕΙΑΚΗ ΥΠΗΡΕΣΙΑ: len=2, ['ΟΙΚΟΝΟΜΙΚΗ ΔΙΟΙΚΗΣΗ', 'ΕΙΣΠΡΑΞΗ ΔΗΜΟΣΙΩΝ ΕΣΟΔΩΝ']
14. ΠΡΟΣΩΠΙΚΟ: len=4, ['ΥΠΟΥΡΓΕΙΟ ΕΞΩΤΕΡΙΚΩΝ', 'ΥΠΟΥΡΓΕΙΟ ΔΙΚΑΙΟΣΥΝΗΣ', 'ΑΝΩΤΑΤΗ ΣΧΟΛΗ ΚΑΛΩΝ ΤΕΧΝΩΝ', 'ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΟΝΙΚΗΣ']
15. ΕΞΕΤΑΣΕΙΣ ΓΙΑ ΤΗΝ ΠΡΟΣΛΗΨΗ ΠΡΟΣΩΠΙΚΟΥ: len=2, ['ΥΠΟΥΡΓΕΙΟ ΟΙΚΟΝΟΜΙΚΩΝ', 'ΥΠΟΥΡΓΕΙΟ ΕΘΝΙΚΗΣ ΟΙΚΟΝΟΜΙΑΣ']
16. ΔΙΑΦΟΡΕΣ ΔΙΑΤΑΞΕΙΣ: len=7, ['ΔΗΜΟΣΙΑ ΚΤΗΜΑΤΑ', 'ΑΝΩΤΑΤΗ ΕΚΠΑΙΔΕΥΣΗ', 'ΙΔΙΩΤΙΚΟ ΝΑΥΤΙΚΟ ΔΙΚΑΙΟ', 'ΑΕΡΟΛΙΜΕΝΕΣ', 'ΣΙΔΗΡΟΔΡΟΜΟΙ ΓΕΝΙΚΑ', 'ΕΚΜΕΤΑΛΛΕΥΣΗ ΘΑΛΑΣΣΙΩΝ ΣΥΓΚΟΙΝΩΝΙΩΝ', 'ΔΙΑΦΟΡΟΙ ΤΕΛΩΝΕΙΑΚΟΙ ΝΟΜΟΙ']
17. ΤΕΧΝΙΚΗ ΥΠΗΡΕΣΙΑ: len=2, ['ΥΠΟΥΡΓΕΙΟΔΙΚΑΙΟΣΥΝΗΣ', 'ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ'] ([60, 192]) 18) ΘΕΟΛΟΓΙΚΗ ΣΧΟΛΗ[647]: len=2, ['ΣΧΟΛΕΣ ΠΑΝΕΠΙΣΤΗΜΙΟΥ ΑΘΗΝΩΝ', 'ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΟΝΙΚΗΣ']
18. ΔΙΑΦΟΡΟΙ ΝΟΜΟΙ: len=5, ['ΥΠΟΥΡΓΕΙΟ ΕΞΩΤΕΡΙΚΩΝ', 'ΟΙΚΟΝΟΜΙΚΗ ΔΙΟΙΚΗΣΗ', 'ΤΟΠΙΚΗ ΑΥΤΟΔΙΟΙΚΗΣΗ', 'ΔΙΑΦΟΡΑ ΘΕΜΑΤΑ', 'ΔΙΑΦΟΡΑ ΣΤΡΑΤΙΩΤΙΚΑ ΘΕΜΑΤΑ']
19. ΤΕΧΝΙΚΑ ΕΡΓΑ: len=2, ['ΣΧΟΛΕΣ ΠΑΝΕΠΙΣΤΗΜΙΟΥ ΑΘΗΝΩΝ', 'ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΟΝΙΚΗΣ']
20. ΘΕΟΛΟΓΙΚΗ ΣΧΟΛΗ[647]: len=2, ['ΣΧΟΛΕΣ ΠΑΝΕΠΙΣΤΗΜΙΟΥ ΑΘΗΝΩΝ', 'ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΟΝΙΚΗΣ']
21. ΥΠΟΘΗΚΗ: len=3, ['ΔΙΟΙΚΗΣΗ ΠΟΛΙΤΙΚΗΣ ΑΕΡΟΠΟΡΙΑΣ', 'ΙΔΙΩΤΙΚΟ ΝΑΥΤΙΚΟ ΔΙΚΑΙΟ', 'ΕΜΠΡΑΓΜΑΤΟΣ ΑΣΦΑΛΕΙΑ']
22. ΓΕΝΙΚΑ: len=2, ['ΠΝΕΥΜΑΤΙΚΗ ΙΔΙΟΚΤΗΣΙΑ', 'ΑΣΤΥΝΟΜΙΚΟΊ ΣΚΎΛΟΙ']
23. ΕΠΙΔΟΜΑ ΣΤΟΛΗΣ: len=2, ['ΑΠΟΔΟΧΕΣ ΔΗΜΟΣΙΩΝ ΥΠΑΛΛΗΛΩΝ', 'ΑΠΟΔΟΧΕΣ ΣΤΡΑΤΙΩΤΙΚΩΝ']
24. ΚΑΤΑΣΤΑΤΙΚΕΣ ΔΙΑΤΑΞΕΙΣ ΠΑΡΟΧΩΝ: len=2,['ΠΑΡΟΧΟΙ ΣΤΑΘΕΡΩΝ ΗΛΕΚΤΡΟΝΙΚΩΝ ΕΠΙΚΟΙΝΩΝΙΩΝ', 'ΠΑΡΟΧΟΙΚΙΝΗΤΩΝ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ']
25. ΠΕΙΘΑΡΧΙΚΑ ΣΥΜΒΟΥΛΙΑ[1760]: len=2, ['ΠΡΟΣΩΠΙΚΟ ΤΩΝΔΙΚΑΣΤΗΡΙΩΝ', 'ΠΟΙΝΙΚΟ ΚΑΙ ΠΕΙΘΑΡΧΙΚΟ ΔΙΚΑΙΟ']

Total Subjects With Multiple Parents: 25

A.1.2 Circles and Self loops

In Chapter 3 of this thesis, we have meticulously described the ground elements that render the dataset a directed graph and not a tree.A key reason is that

Raptarchis47k contains *"self-loops"*[1] and *"circles"* [2] which disturbs its expected structure. In fact, the truth is that we can discern the following cases:

1. Labels with the same names on *Layers [1, 2, 3]*
2. Labels with the sames names on *Layers [1, 2]*
3. Labels with the sames names on *Layers [1, 3]*
4. Labels with the sames names on *Layers [2, 3]*

So, for *Case-1* we have:

1. ΟΙΚΟΝΟΜΙΚΗ ΔΙΟΙΚΗΣΗ ==> ΟΙΚΟΝΟΜΙΚΗ ΔΙΟΙΚΗΣΗ ==>ΟΙΚΟΝΟΜΙΚΗ ΔΙΟΙΚΗΣΗ

For *Case-2* we have:

1. ΟΙΚΟΝΟΜΙΚΗ ΔΙΟΙΚΗΣΗ ==> ΟΙΚΟΝΟΜΙΚΗ ΔΙΟΙΚΗΣΗ *(Cross-parent [3])*
2. ΔΗΜΟΣΙΟ ΛΟΓΙΣΤΙΚΟ ==> ΔΗΜΟΣΙΟ ΛΟΓΙΣΤΙΚΟ *(Direct-parent [4])*
3. ΝΟΜΙΚΑ ΠΡΟΣΩΠΑ ΔΗΜΟΣΙΟΥ ΔΙΚΑΙΟΥ ==> ΝΟΜΙΚΑ ΠΡΟΣΩΠΑ ΔΗΜΟΣΙΟΥ ΔΙΚΑΙΟΥ *(Direct-parent)*
4. ΠΟΛΙΤΙΚΗ ΔΙΚΟΝΟΜΙΑ ==> ΠΟΛΙΤΙΚΗ ΔΙΚΟΝΟΜΙΑ *(Direct-parent)*

For *Case-3* we have:

1. ΑΓΟΡΑΝΟΜΙΚΗ ΝΟΜΟΘΕΣΙΑ ==> ['ΔΙΟΙΚΗΣΗ ΕΦΟΔΙΑΣΜΟΥ'] ==> ΑΓΟΡΑΝΟΜΙΚΗ ΝΟΜΟΘΕΣΙΑ
2. ΟΙΚΟΝΟΜΙΚΗ ΔΙΟΙΚΗΣΗ ==> ['ΣΧΟΛΕΣ ΠΑΝΕΠΙΣΤΗΜΙΟΥ ΑΘΗΝΩΝ'] ==> ΟΙΚΟΝΟΜΙΚΗ ΔΙΟΙΚΗΣΗ

Ultimately, for *Case-4* we have:

1. ΔΙΑΦΟΡΑ ==> ΔΙΑΦΟΡΑ *(Direct-parent)*
2. ΠΟΛΕΜΙΚΗ ΔΙΑΘΕΣΙΜΟΤΗΤΑ ==> ΠΟΛΕΜΙΚΗ ΔΙΑΘΕΣΙΜΟΤΗΤΑ *(Direct-parent)*
3. ΔΙΕΘΝΕΙΣ ΣΥΜΒΑΣΕΙΣ ==> ΔΙΕΘΝΕΙΣ ΣΥΜΒΑΣΕΙΣ *(Cross-parent)*
4. ΓΕΝΙΚΟ ΧΗΜΕΙΟ ΤΟΥ ΚΡΑΤΟΥΣ ==> ΓΕΝΙΚΟ ΧΗΜΕΙΟ ΤΟΥ ΚΡΑΤΟΥΣ *(Direct-parent)*
5. ΑΣΤΙΚΟΣ ΚΩΔΙΚΑΣ ==> ΑΣΤΙΚΟΣ ΚΩΔΙΚΑΣ *(Direct-parent)*

[1] We mention as self-loops to all the vertexes of a layer L_i for which exist at least one child-node with the same name on L_{i+1}.

[2] We mention as circles all the trails/paths that connect a vertex of a layer L_i to another child-nodes on L_{i+j}/ i, j = {0, 1, 2, 3} and i $\neq j$.

[3] Vertexes that do not belong to the same branch with their descendant nodes.

[4] Vertexes that do belong to the same branch with their descendant nodes.

6. ΔΙΚΑΣΤΙΚΟΙ ΕΠΙΜΕΛΗΤΕΣ ==> ΔΙΚΑΣΤΙΚΟΙ ΕΠΙΜΕΛΗΤΕΣ *(Direct-parent)*
7. ΠΟΛΕΜΙΚΕΣ ΣΥΝΤΑΞΕΙΣ ==> ΠΟΛΕΜΙΚΕΣ ΣΥΝΤΑΞΕΙΣ *(Direct-parent)*
8. ΥΠΟΥΡΓΕΙΟ ΕΘΝΙΚΗΣ ΟΙΚΟΝΟΜΙΑΣ ==> ΥΠΟΥΡΓΕΙΟ ΕΘΝΙΚΗΣ ΟΙΚΟΝΟΜΙΑΣ *(Direct-parent)*
9. ΟΙΚΟΝΟΜΙΚΗ ΔΙΟΙΚΗΣΗ ==> ΟΙΚΟΝΟΜΙΚΗ ΔΙΟΙΚΗΣΗ *(Cross-parent)*
10. ΔΙΑΦΟΡΕΣ ΒΙΟΜΗΧΑΝΙΕΣ ==> ΔΙΑΦΟΡΕΣ ΒΙΟΜΗΧΑΝΙΕΣ *(Direct-parent)*
11. ΒΑΣΙΛΕΙΑ ΚΑΙ ΑΝΤΙΒΑΣΙΛΕΙΑ ==> ΒΑΣΙΛΕΙΑ ΚΑΙ ΑΝΤΙΒΑΣΙΛΕΙΑ *(Direct-parent)*
12. ΑΓΙΟΝ ΟΡΟΣ ==> ΑΓΙΟΝ ΟΡΟΣ *(Direct-parent)*
13. ΕΚΚΛΗΣΙΑ ΚΡΗΤΗΣ ==> ΕΚΚΛΗΣΙΑ ΚΡΗΤΗΣ *(Direct-parent)*
14. ΕΚΚΛΗΣΙΑ ΙΟΝΙΩΝ ΝΗΣΩΝ ==> ΕΚΚΛΗΣΙΑ ΙΟΝΙΩΝ ΝΗΣΩΝ *(Direct-parent)*
15. ΑΝΩΤΑΤΟ ΕΙΔΙΚΟ ΔΙΚΑΣΤΗΡΙΟ ==> ΑΝΩΤΑΤΟ ΕΙΔΙΚΟ ΔΙΚΑΣΤΗΡΙΟ *(Direct-parent)*
16. ΒΑΣΙΛΙΚΑ ΙΔΡΥΜΑΤΑ ==> ΒΑΣΙΛΙΚΑ ΙΔΡΥΜΑΤΑ *(Direct-parent)*
17. ΣΩΜΑΤΙΚΗ ΑΓΩΓΗ ==> ΣΩΜΑΤΙΚΗ ΑΓΩΓΗ *(Cross-parent)*
18. ΚΑΠΝΟΣ ==> ΚΑΠΝΟΣ *(Direct-parent)*
19. ΕΚΤΕΛΕΣΗ[356] ==> ΕΚΤΕΛΕΣΗ *(Cross-parent)*

A.2 Data Analysis

The Raptarchis47k dataset is a data collection that contains Greek legislation from the period between 1832 to 2015. This wide range of time however displays a lot of interest since reflecting the historical course and the genuine evolution of the actual legal code of Greece. In view of this, it is necessary, for the better understanding and comprehension of the corpus, to find and reveal some of its hidden and most profound parameters. For this purpose, we will proceed with a brief but concise analysis of the contextual information that is contained in our data collection.

First of all, we start by presenting the *word cloud* of the dataset in Figure A.1. Broadly speaking, *word or tag could* is a convenient visual representation method that is used to depict the most significant and noteworthy keywords in a set of textual data. By observing it we are able to immediately perceive and distinguish the prominent terms from the less important and trivial ones. So, in our case, we can easily figure out and realize the nature of our collection from words like *διάταγμα, διατάεισ, etc*.

Figure A.1: The World Cloud of the dataset.

On Figure A.2 we can see the total number of legal resources that were produced annually from zero (1832) to two-hundred-three (2015). From this image, we notice the constant reduction of the legislation which practically implies the steady abandonment and termination of the Raptarchis project.

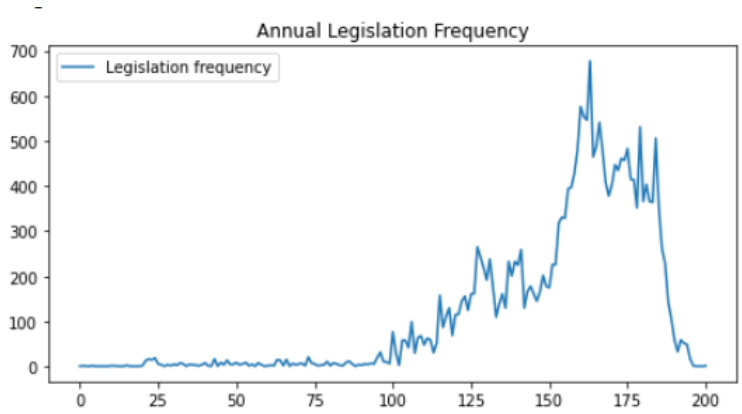


Figure A.2: The annual legislation of Greece with zero to stand for 1832 and 203 for 2015 respectively.

Moving forward, the Figure A.3 expresses the distribution of tokens per sentence in the dataset. This diagram is a trustworthy advisor and consultant that reliably suggests how many tokens are sufficient for the development and the training of our models.

Finally, on Figure A.4, Figure A.5, and Figure A.6 we have the overall number of instances for the *Volume, Chapter,* and *Subject* category. On these figures, it is readily discernible that as we go from the more general (upper) to the more specific (lower) layers of the hierarchy the real number of samples is reduced while the magnitude of concepts increases rapidly. This fact is in accordance with our intuition while at the same time signifies the incrementing difficulty of the task as we engage with the lower levels of the label graph. Another observation is that the instance distributions of the classes are uneven and thus it is meaningful to discriminate them into Frequent/Not-Frequent clusters.

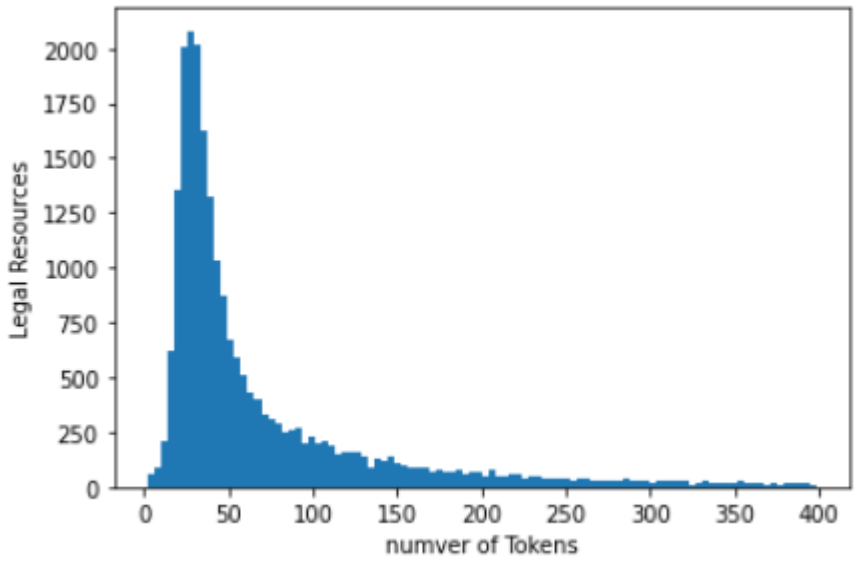


Figure A.3: The number of tokens per document.

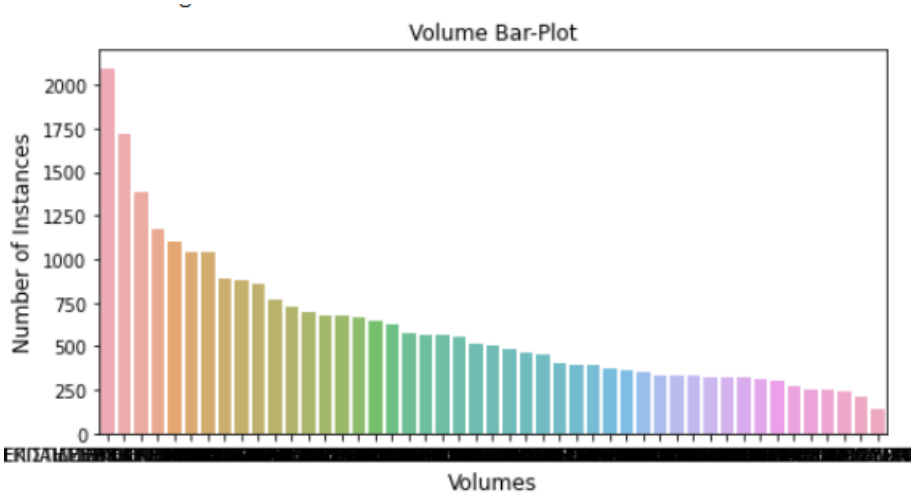


Figure A.4: The Frequency for Volume labels.

A.3 Experiments

A.3.1 [CLS]-Representations

Supplementary to the results of Section 5.2 and in order to unfold the full behavior of our models we considered it appropriate to provide the full findings that were emerged from the training process of Chapter 4. Thus, the Fig-Figure A.7 displays the [CLS] angular distances across the BERT layers for the five layer-wise guided models of Subsection 4.2.1. Additionally, the Fig-Figure A.8 exhibits the corresponding results exactly as they are given in the work of Manginas et al. [2020] for the development set of MIMIC-III [6].

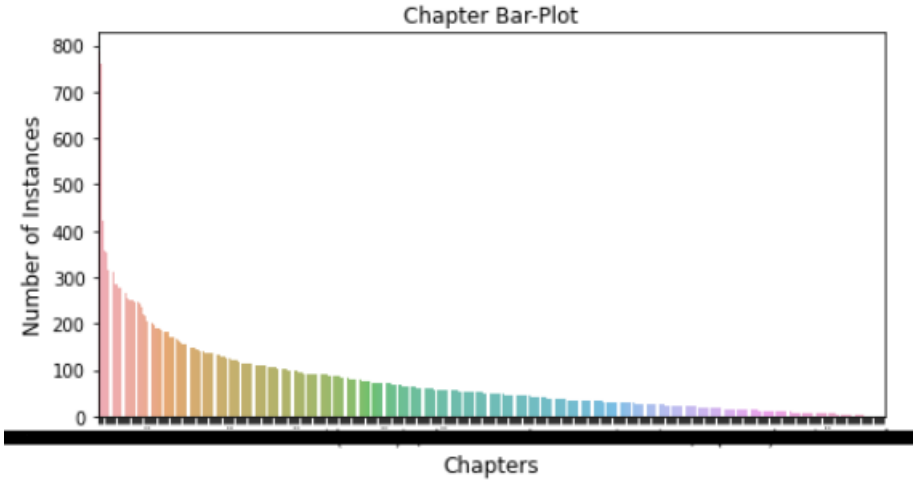


Figure A.5: The Frequency for Chapter labels.

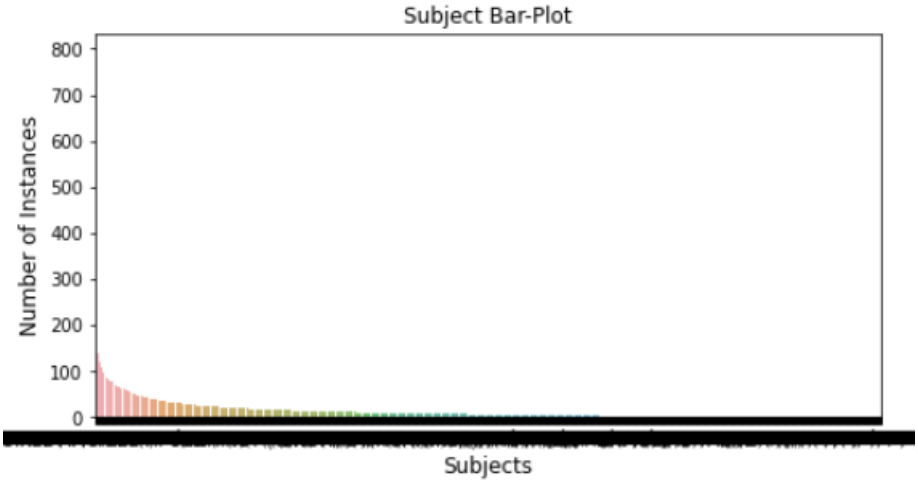


Figure A.6: The Frequency for Subject labels.

A.3.2 Other results

At last but not least, the *Fig-Figure A.9, Fig-Figure A.10, and Fig-Figure A.11* provide some extra results relevant to the performance of the *Greek-Hybrid* model.

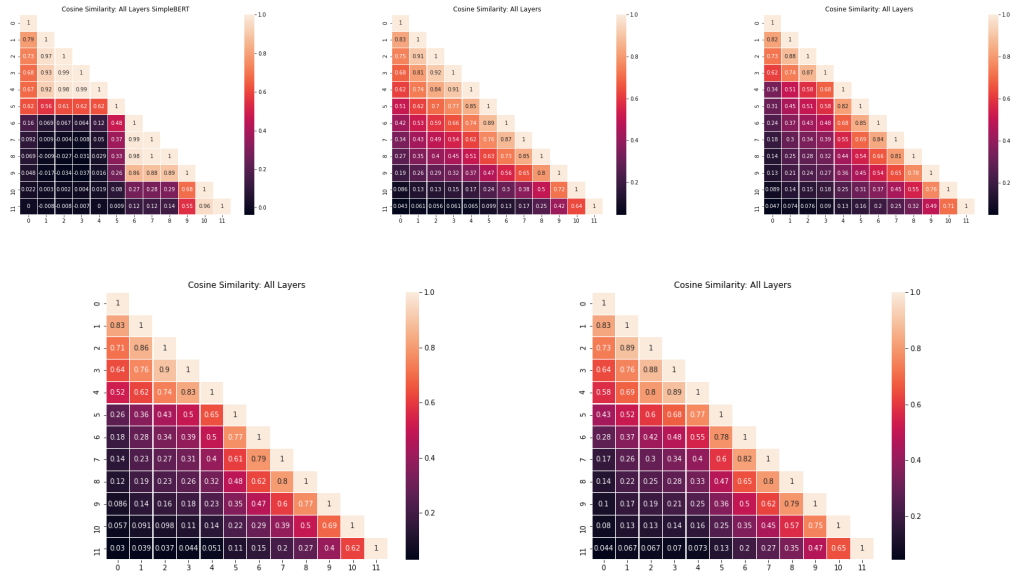


Figure A.7: The [CLS] cosine similarity for the multi-lingual case of the five models of Fig- 4.1a.

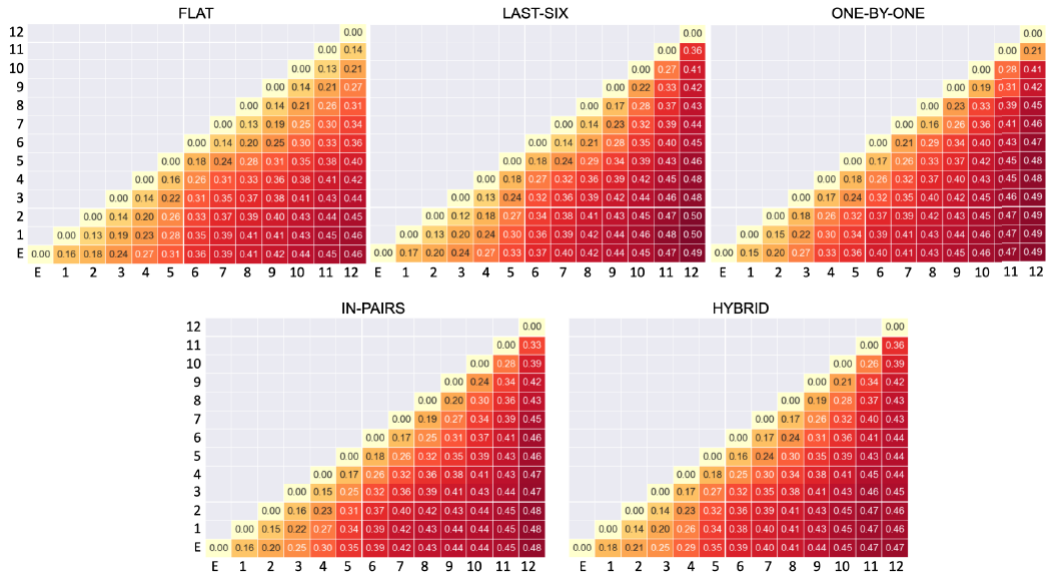


Figure A.8: The [CLS] angular distances across the BERT layers for the developemnt set of MIMIC-III [6] dataset [5].

ΕΘΝΙΚΗ ΑΜΥΝΑ	0.68	0.86	0.76	203
ΔΗΜΟΣΙΑ ΕΡΓΑ	0.91	0.89	0.90	247
ΚΟΙΝΩΝΙΚΗ ΠΡΟΝΟΙΑ	0.80	0.91	0.85	144
ΕΘΝΙΚΗ ΟΙΚΟΝΟΜΙΑ	0.86	0.76	0.81	91
ΠΟΛΕΜΙΚΗ ΑΕΡΟΠΟΡΙΑ	0.74	0.82	0.78	49
ΕΜΜΕΣΗ ΦΟΡΟΛΟΓΙΑ	0.80	0.90	0.85	123
ΔΑΣΗ ΚΑΙ ΚΤΗΝΟΤΡΟΦΙΑ	0.94	0.83	0.88	71
ΠΟΛΙΤΙΚΗ ΑΕΡΟΠΟΡΙΑ	0.91	0.95	0.93	63
ΕΚΚΛΗΣΙΑΣΤΙΚΗ ΝΟΜΟΘΕΣΙΑ	0.91	0.92	0.92	102
ΑΜΕΣΗ ΦΟΡΟΛΟΓΙΑ	0.90	0.83	0.86	177
ΕΛΕΓΚΤΙΚΟ ΣΥΝΕΔΡΙΟ ΚΑΙ ΣΥΝΤΑΞΕΙΣ	0.87	0.76	0.81	87
ΝΟΜΟΘΕΣΙΑ ΔΗΜΩΝ ΚΑΙ ΚΟΙΝΟΤΗΤΩΝ	0.93	0.80	0.86	133
ΠΕΡΙΟΥΣΙΑ ΔΗΜΟΣΙΟΥ ΚΑΙ ΝΟΜΙΣΜΑ	0.92	0.72	0.81	94
ΤΕΛΩΝΕΙΑΚΗ ΝΟΜΟΘΕΣΙΑ	0.85	0.84	0.85	125
ΑΣΤΙΚΗ ΝΟΜΟΘΕΣΙΑ	0.85	0.90	0.88	90
ΔΗΜΟΣΙΟ ΛΟΓΙΣΤΙΚΟ	0.83	0.74	0.78	107
ΕΜΠΟΡΙΚΗ ΝΟΜΟΘΕΣΙΑ	0.87	0.78	0.82	67
ΠΟΛΕΜΙΚΟ ΝΑΥΤΙΚΟ	0.96	0.79	0.87	62
ΑΓΡΟΤΙΚΗ ΝΟΜΟΘΕΣΙΑ	0.93	0.82	0.87	50
ΠΟΛΙΤΙΚΗ ΔΙΚΟΝΟΜΙΑ	0.94	0.56	0.70	27
accuracy			0.89	7612
macro avg	0.89	0.86	0.87	7612
weighted avg	0.89	0.89	0.89	7612

Figure A.9: The *Precision, Recall, and F1-score* for various Volume categories of *Greek-Hybrid-Freq10* model.

	precision	recall	f1-score	support
ΠΡΟΣΤΑΣΙΑ ΚΑΙ ΚΙΝΗΤΡΑ ΙΔΙΩΤΙΚΩΝ ΕΠΕΝΔΥΣΕΩΝ	0.51	0.69	0.58	45
ΤΥΠΟΣ	0.74	0.82	0.78	17
ΥΠΟΥΡΓΕΙΟ ΕΣΩΤΕΡΙΚΩΝ ΔΗΜ.ΔΙΟΙΚΗΣΗΣ ΚΑΙ ΑΠΟΚΕΝΤΡΩΣΗΣ	0.91	0.91	0.91	34
ΛΙΜΕΝΕΣ	0.90	0.95	0.92	78
ΟΡΓΑΝΙΣΜΟΣ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ ΕΛΛΑΔΑΣ (Ο.Τ.Ε.)	0.80	0.88	0.83	40
ΑΡΤΟΣ	0.89	0.81	0.85	21
ΕΠΑΓΓΕΛΜΑΤΙΚΗ ΕΚΠΑΙΔΕΥΣΗ	0.85	0.90	0.87	134
ΤΡΑΠΕΖΕΣ	0.84	0.90	0.87	79
ΠΑΝΕΠΙΣΤΗΜΙΟ ΙΩΑΝΝΙΝΩΝ	0.94	0.82	0.88	82
ΑΣΦΑΛΙΣΤΙΚΑ ΤΑΜΕΙΑ ΤΥΠΟΥ	1.00	0.95	0.98	43
ΚΡΑΤΙΚΗ	0.94	1.00	0.97	34
ΠΡΟΣΩΠΙΚΟ ΤΩΝ ΔΙΚΑΣΤΗΡΙΩΝ	0.90	0.65	0.76	43
ΑΥΤΟΚΙΝΗΤΑ	0.88	0.92	0.90	249
ΕΝΙΣΧΥΣΙΣ ΤΗΣ ΓΕΩΡΓΙΑΣ	0.67	0.73	0.70	30
ΥΓΕΙΟΝΟΜΙΚΗ ΑΝΤΙΛΗΨΗ	0.76	0.87	0.81	68
ΚΑΤΑΣΤΑΣΗ ΔΗΜΟΣΙΩΝ ΥΠΑΛΛΗΛΩΝ	0.69	0.74	0.71	69
ΑΣΥΡΜΑΤΟΣ	0.93	0.88	0.90	16
ΔΙΟΙΚΗΣΗ ΕΜΠΟΡΙΚΟΥ ΝΑΥΤΙΚΟΥ	0.69	0.86	0.77	21
ΣΩΜΑΤΙΚΗ ΑΓΩΓΗ	0.94	0.91	0.92	64
ΤΑΜΕΙΟ ΣΥΝΤΑΞΕΩΝ ΝΟΜΙΚΩΝ	1.00	1.00	1.00	16
ΣΥΜΒΟΛΑΙΟΓΡΑΦΟΙ	0.97	1.00	0.98	32
ΣΤΟΙΧΕΙΩΔΗΣ ΕΚΠΑΙΔΕΥΣΗ	0.75	0.92	0.82	61
ΠΡΟΝΟΙΑ ΠΛΗΡΩΜΑΤΩΝ Ε.Ν	0.90	1.00	0.95	18
ΑΤΕΘΝΕΙΣ ΜΕΤΑΦΟΡΕΣ	0.74	0.87	0.80	30

✓ 0s completed at 11:53 AM

Figure A.10: The *Precision, Recall, and F1-score* for various Chapter categories of *Greek-Hybrid-Freq10* model.

	precision	recall	f1-score	support
ΟΡΟΙ - ΠΡΟΔΙΑΓΡΑΦΕΣ ΤΥΠΟΠΟΙΗΣΗΣ	0.68	0.79	0.73	19
ΔΗΜΟΣΙΟΓΡΑΦΙΚΟΣ ΧΑΡΤΗΣ	1.00	0.78	0.88	9
ΟΡΓΑΝΙΣΜΟΣ	0.63	0.69	0.66	32
ΛΙΜΕΝΙΚΑ ΤΑΜΕΙΑ – ΛΙΜΕΝΙΚΑ ΕΡΓΑ	0.64	0.75	0.69	12
ΤΙΜΟΛΟΓΙΑ Ο.Τ.Ε - ΚΟΣΤΟΛΟΓΗΣΗ ΥΠΗΡΕΣΙΩΝ Ο.Τ.Ε	0.59	1.00	0.74	24
ΤΑΜΕΙΟ ΕΠΙΚΟΥΡΙΚΗΣ ΑΣΦΑΛΙΣΗΣ ΑΡΤΟΠΟΙΩΝ	1.00	0.78	0.88	9
ΣΙΒΙΤΑΝΙΔΕΙΟΣ ΣΧΟΛΗ	1.00	0.89	0.94	9
ΕΘΝΙΚΗ ΤΡΑΠΕΖΑ ΤΗΣ ΕΛΛΑΔΟΣ	0.67	0.50	0.57	4
ΠΑΝΕΠΙΣΤΗΜΙΑ ΑΙΓΑΙΟΥ, ΙΟΝΙΟΥ ΚΑΙ ΘΕΣΣΑΛΙΑΣ	0.85	0.79	0.81	14
ΤΑΜΕΙΟ ΑΣΦΑΛΙΣΕΩΣ ΙΔΙΟΚΤΗΤΩΝ	0.88	1.00	0.93	14
ΓΕΝΙΚΕΣ ΔΙΑΤΑΞΕΙΣ ΓΙΑ ΡΑΔΙΟΦΩΝΙΑ – ΤΗΛΕΟΡΑΣΗ	0.81	1.00	0.90	30
ΔΙΚΑΣΤΙΚΟΙ ΛΕΙΤΟΥΡΓΟΙ - ΕΘΝΙΚΗ ΣΧΟΛΗ ΔΙΚΑΣΤΩΝ	0.87	0.61	0.71	33
ΦΟΡΤΗΓΑ ΑΥΤΟΚΙΝΗΤΑ	0.95	0.70	0.80	50
ΠΡΟΩΘΗΣΗ ΓΕΩΡΓΙΚΗΣ ΠΑΡΑΓΩΓΗΣ-ΕΘ.Ι.ΑΓ.Ε	1.00	0.09	0.17	11
ΝΟΣΗΛΕΥΤΙΚΑ ΙΔΡΥΜΑΤΑ	0.45	0.33	0.38	15
ΠΡΟΣΛΗΨΕΙΣ ΣΤΟ ΔΗΜΟΣΙΟ	0.43	0.90	0.58	10
[ΔΙΩΤΙΚΟΙ ΣΤΑΘΜΟΙ ΑΣΥΡΜΑΤΟΥ - ΧΡΗΣΗ ΡΑΔΙΟΣΥΧΝΟΤΗΤΩΝ	1.00	0.88	0.93	16
ΥΠΟΥΡΓΕΙΟ ΕΜΠΟΡΙΚΗΣ ΝΑΥΤΙΛΙΑΣ	0.69	0.90	0.78	20
ΕΞΩΣΧΟΛΙΚΗ ΣΩΜΑΤΙΚΗ ΑΓΩΓΗ	0.88	0.94	0.91	32
ΚΛΑΔΟΣ ΕΠΙΚΟΥΡΙΚΗΣ ΑΣΦΑΛΙΣΕΩΣ ΔΙΚΗΓΟΡΩΝ (Κ.Ε.Α.Δ.)	0.90	1.00	0.95	9
ΣΥΜΒΟΛΑΙΟΓΡΑΦΙΚΑ ΔΙΚΑΙΩΜΑΤΑ	0.00	0.00	0.00	7
ΓΕΝΙΚΑ ΠΕΡΙ ΚΥΚΛΟΦΟΡΙΑΣ ΑΥΤΟΚΙΝΗΤΩΝ	0.57	0.78	0.66	54
ΛΕΙΤΟΥΡΓΟΙ ΣΤΟΙΧΕΙΩΔΟΥΣ	0.56	0.56	0.56	9
ΕΣΤΙΑ ΝΑΥΤΙΚΩΝ	1.00	0.50	0.67	2

Figure A.11: The *Precision, Recall, and F1-score* for various Subject categories of *Greek-Hybrid-Freq10* model.

BIBLIOGRAPHY

[1] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[2] Christos N. Papaloukas. *Legal Text Classification based on Greek Legislation*. Dissertation. NKUA Pergamos, 2020.

[3] J. Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, 2019.

[4] Nikolaos Manginas, Ilias Chalkidis, and Prodromos Malakasiotis. Layer-wise guided training for bert: Learning incrementally refined document representations. In *SPNLP*, 2020.

[5] Nikolaos Manginas, Ilias Chalkidis, and Prodromos Malakasiotis. Layer-wise guided training for bert: Learning incrementally refined document representations. *arXiv preprint arXiv:2010.05763*, 2020.

[6] Alistair E. W. Johnson, T. Pollard, Lu Shen, Li wei H. Lehman, M. Feng, M. Ghassemi, Benjamin Moody, Peter Szolovits, L. Celi, and R. Mark. Mimic-iii, a freely accessible critical care database. *Scientific Data*, 3, 2016.

[7] Yoon Kim. Convolutional neural networks for sentence classification. In *EMNLP*, 2014.

[8] Kyunghyun Cho, B. V. Merrienboer, Çaglar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *EMNLP*, 2014.

[9] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9: 1735–1780, 1997.

[10] Ashish Vaswani, Noam M. Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *ArXiv*, abs/1706.03762, 2017.

[11] A. Nazarenko and A. Wyner. Legal nlp introduction. 2018.

[12] An artificial intelligence based analysis in legal domain. 2020.

[13] Haoxi Zhong, Chaojun Xiao, Cunchao Tu, Tianyang Zhang, Zhiyuan Liu, and Maosong Sun. How does nlp benefit legal system: A summary of legal artificial intelligence. In *ACL*, 2020.

[14] Ilias Chalkidis, Manos Fergadiotis, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos. Legal-bert: The muppets straight out of law school. *ArXiv*, abs/2010.02559, 2020.

[15] Haoxi Zhong, Zhipeng Guo, Cunchao Tu, Chaojun Xiao, Zhiyuan Liu, and Maosong Sun. Legal judgment prediction via topological learning. In *EMNLP*, 2018.

[16] X. Duan, Baoxin Wang, Ziyue Wang, Wentao Ma, Yiming Cui, Dayong Wu, Shijin Wang, Ting Liu, Tianxiang Huo, Zhen Hu, Heng Wang, and Zhiyuan Liu. Cjrc: A reliable human-annotated benchmark dataset for chinese judicial reading comprehension. *ArXiv*, abs/1912.09156, 2019.

[17] Ilias Chalkidis, Manos Fergadiotis, Prodromos Malakasiotis, and Ion Androutsopoulos. Large-scale multi-label text classification on eu legislation. In *ACL*, 2019.

[18] Don Tuggener, Pius von Däniken, Thomas Peetz, and Mark Cieliebak. Ledgar: A large-scale multi-label corpus for text classification of legal provisions in contracts. In *LREC*, 2020.

[19] Min-Ling Zhang and Z. Zhou. A review on multi-label learning algorithms. *IEEE Transactions on Knowledge and Data Engineering*, 26:1819–1837, 2014.

[20] Wikipedia. Multi-label classification @ONLINE. https://en.wikipedia.org/wiki/Multi-label_classification, .

[21] E. A. Cherman, M. C. Monard, and Jean Metz. Multi-label problem transformation methods: a case study. *CLEI Electron. J.*, 14, 2011.

[22] Min-Ling Zhang, Yukun Li, Xu-Ying Liu, and Xin Geng. Binary relevance for multi-label learning: an overview. *Frontiers of Computer Science*, 12:191–202, 2017.

[23] J. Read, B. Pfahringer, G. Holmes, and Eibe Frank. Classifier chains for multi-label classification. *Machine Learning*, 85:333–359, 2011.

[24] Grigorios Tsoumakas, I. Katakis, and I. Vlahavas. Random k-labelsets for multilabel classification. *IEEE Transactions on Knowledge and Data Engineering*, 23:1079–1089, 2011.

[25] Grigorios Tsoumakas and I. Vlahavas. Random k -labelsets: An ensemble method for multilabel classification. In *ECML*, 2007.

[26] Y. Hu and Jenq-Neng Hwang. Committee machines. In *Encyclopedia of Machine Learning and Data Mining*, 2017.

[27] Min-Ling Zhang and Z. Zhou. Ml-knn: A lazy learning approach to multi-label learning. *Pattern Recognit.*, 40:2038–2048, 2007.

[28] Shihchieh Chou and Chang-Ling Hsu. Mmdt: a multi-valued and multi-labeled decision tree classifier for data mining. *Expert Syst. Appl.*, 28:799–812, 2005.

[29] Yen-Liang Chen, Chang-Ling Hsu, and Shihchieh Chou. Constructing a multi-valued and multi-labeled decision tree. *Expert Syst. Appl.*, 25:199–209, 2003.

[30] Min-Ling Zhang and Z. Zhou. Multilabel neural networks with applications to functional genomics and text categorization. *IEEE Transactions on Knowledge and Data Engineering*, 18:1338–1351, 2006.

[31] Rohit Babbar and B. Schölkopf. Dismec: Distributed sparse machines for extreme multi-label classification. *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, 2017.

[32] I. E. Yen, Xiangru Huang, Wei Dai, Pradeep Ravikumar, I. Dhillon, and E. Xing. Ppdsparse: A parallel primal-dual sparse method for extreme classification. *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2017.

[33] Rohit Babbar and B. Schölkopf. Data scarcity, robustness and extreme multi-label classification. *Machine Learning*, pages 1–23, 2019.

[34] K. Bhatia, Himanshu Jain, Purushottam Kar, M. Varma, and Prateek Jain. Sparse local embeddings for extreme multi-label classification. In *NIPS*, 2015.

[35] Yukihiro Tagami. Annexml: Approximate nearest neighbor search for extreme multi-label classification. *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2017.

[36] Sujay Khandagale, Han Xiao, and Rohit Babbar. Bonsai: diverse and shallow trees for extreme multi-label classification. *Machine Learning*, pages 1 – 21, 2020.

[37] Jiaqi Lu, Jun Zheng, and Wen xin Hu. Pparabel: Parallel partitioned label trees for extreme classification. In *NPC*, 2019.

[38] Wei-Cheng Chang, Hsiang-Fu Yu, Kai Zhong, Yiming Yang, and I. Dhillon. X-bert: extreme multi-label text classification with bert. 2019.

[39] Indexer, Qiao Jin, Bhuwan Dhingra, and William W. Cohen. Attentionmesh: Simple, effective and interpretable automatic mesh indexer. 2018.

[40] R. You, Zihan Zhang, Ziye Wang, Suyang Dai, Hiroshi Mamitsuka, and Shanfeng Zhu. Attentionxml: Label tree-based attention-aware deep model for high-performance extreme multi-label text classification. In *NeurIPS*, 2019.

[41] Jingzhou Liu, Wei-Cheng Chang, Yuexin Wu, and Yiming Yang. Deep learning for extreme multi-label text classification. *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2017.

[42] Guangxu Xun, Kishlay Jha, Jianhui Sun, and Aidong Zhang. Correlation networks for extreme multi-label text classification. *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2020.

[43] Julian McAuley and J. Leskovec. Hidden factors and hidden topics: understanding rating dimensions with review text. *Proceedings of the 7th ACM conference on Recommender systems*, 2013.

[44] K. Bhatia, K. Dahiya, H. Jain, P. Kar, A. Mittal, Y. Prabhu, and M. Varma. The extreme classification repository: Multi-label datasets and code, 2016. URL http://manikvarma.org/downloads/XC/XMLRepository.html.

[45] Yu Zhang, Zhihong Shen, Yuxiao Dong, Kuansan Wang, and Jiawei Han. Match: Metadata-aware text classification in a large hierarchy. *Proceedings of the Web Conference 2021*, 2021.

[46] Kuansan Wang, Zhihong Shen, Chiyuan Huang, Chieh-Han Wu, Yuxiao Dong, and Anshul Kanakia. Microsoft academic graph: When experts are not enough. *Quantitative Science Studies*, 1:396–413, 2020.

[47] Zhiyong Lu. Pubmed and beyond: a survey of web tools for searching biomedical literature. *Database: The Journal of Biological Databases and Curation*, 2011, 2011.

[48] A. Zubiaga. Enhancing navigation on wikipedia with social tags. *ArXiv*, abs/1202.5469, 2012.

[49] Wei-Cheng Chang, Hsiang-Fu Yu, Kai Zhong, Yiming Yang, and I. Dhillon. Taming pretrained transformers for extreme multi-label text classification. *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2020.

[50] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. *ArXiv*, abs/1804.07461, 2018.

[51] Sashank J. Reddi, Satyen Kale, F. Yu, D. Holtmann-Rice, Jiecao Chen, and Sanjiv Kumar. Stochastic negative mining for learning with large output spaces. In *AISTATS*, 2019.

[52] R. Wetzker, Carsten Zimmermann, and C. Bauckhage. Analyzing social bookmarking systems : A del . icio . us cookbook. 2008.

[53] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, M. Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *ArXiv*, abs/1907.11692, 2019.

[54] Zhilin Yang, Zihang Dai, Yiming Yang, J. Carbonell, R. Salakhutdinov, and Quoc V. Le. Xlnet: Generalized autoregressive pretraining for language understanding. In *NeurIPS*, 2019.

[55] N. Houlsby, A. Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and S. Gelly. Parameter-efficient transfer learning for nlp. In *ICML*, 2019.

[56] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100,000+ questions for machine comprehension of text. In *EMNLP*, 2016.

[57] Ken Lang. Newsweeder: Learning to filter netnews. In Armand Prieditis and Stuart Russell, editors, *Machine Learning Proceedings 1995*, pages 331–339, San Francisco (CA), 1995. Morgan Kaufmann. ISBN 978-1-55860-377-6. doi: https://doi.org/10.1016/B978-1-55860-377-6.50048-7. URL https://www.sciencedirect.com/science/article/pii/B9781558603776500487.

[58] A. Asuncion. Uci machine learning repository, university of california, irvine, school of information and computer sciences. 2007.

[59] Proceedings of the 2011 acm symposium on document engineering, mountain view, ca, usa, september 19-22, 2011. In *ACM Symposium on Document Engineering*, 2011.

[60] Alex Warstadt, Amanpreet Singh, and Samuel R. Bowman. Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics*, 7:625–641, 2019.

[61] Wikipedia. Wiki: Greek government-debt crisis @ONLINE. https://en.wikipedia.org/wiki/Omnibus_bill, June 2009.

[62] Wikipedia. Wiki: Greek government-debt crisi @ONLINE. https://en.wikipedia.org/wiki/Greek_government-debt_crisis, .

[63] Ilias Chalkidis, Manos Fergadiotis, Sotiris Kotitsas, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos. An empirical study on large-scale multi-label text classification including few and zero-shot labels. In *EMNLP*, 2020.

[64] Anthony Rios and Ramakanth Kavuluru. Few-shot and zero-shot multi-label learning for structured label spaces. *Proceedings of the Conference on Empirical Methods in Natural Language Processing. Conference on Empirical Methods in Natural Language Processing*, 2018:3132–3142, 2018.

[65] Trapit Bansal, R. Jha, and A. McCallum. Learning to few-shot learn across diverse natural language classification tasks. In *COLING*, 2020.

[66] Pushpankar Kumar Pushp and Muktabh Mayank Srivastava. Train once, test anywhere: Zero-shot learning for text classification. *ArXiv*, abs/1712.05972, 2017.

[67] Zhiquan Ye, Yuxia Geng, Jiaoyan Chen, Jingmin Chen, Xiaoxiao Xu, Suhang Zheng, F. Wang, J. Zhang, and Huajun Chen. Zero-shot text classification via reinforced self-training. In *ACL*, 2020.

[68] Tomas Mikolov, Ilya Sutskever, Kai Chen, G. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *NIPS*, 2013.

[69] Tomas Mikolov, Kai Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. In *ICLR*, 2013.

[70] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146, 2017.

[71] Jeffrey Pennington, R. Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *EMNLP*, 2014.

[72] Kyunghyun Cho, B. V. Merrienboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder–decoder approaches. In *SSST@EMNLP*, 2014.

[73] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *NAACL*, 2018.

[74] Jeremy Howard and Sebastian Ruder. Universal language model fine-tuning for text classification. In *ACL*, 2018.

[75] Alec Radford, Jeff Wu, R. Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019.

[76] Guillaume Lample and Alexis Conneau. Cross-lingual language model pretraining. In *NeurIPS*, 2019.

[77] John Koutsikakis, Ilias Chalkidis, Prodromos Malakasiotis, and Ion Androutsopoulos. Greek-bert: The greeks visiting sesame street. In *11th Hellenic Conference on Artificial Intelligence*, pages 110–117, 2020.

[78] Aris Kosmopoulos, Ioannis Partalas, Éric Gaussier, G. Paliouras, and Ion Androutsopoulos. Evaluation measures for hierarchical classification: a unified view and novel approaches. *Data Mining and Knowledge Discovery*, 29:820–865, 2014.