Layer-wise Guided Training for BERT:
Learning Incrementally Refined Document Representations

Nikolaos Manginas [†]   **Ilias Chalkidis** [†‡]   **Prodromos Malakasiotis** [†‡]

[†] Institute of Informatics & Telecommunications, NCSR "Demokritos"
[‡] Department of Informatics, Athens University of Economics and Business
[nmanginas,ichalkidis,pmalakasiotis]@iit.demokritos.gr

Abstract

Although BERT is widely used by the NLP community, little is known about its inner workings. Several attempts have been made to shed light on certain aspects of BERT, often with contradicting conclusions. A much raised concern focuses on BERT's over-parameterization and under-utilization issues. To this end, we propose o novel approach to fine-tune BERT in a structured manner. Specifically, we focus on Large Scale Multilabel Text Classification (LMTC) where documents are assigned with one or more labels from a large predefined set of hierarchically organized labels. Our approach guides specific BERT layers to predict labels from specific hierarchy levels. Experimenting with two LMTC datasets we show that this structured fine-tuning approach not only yields better classification results but also leads to better parameter utilization.

1 Introduction

Despite BERT's (Devlin et al., 2019) popularity and effectiveness, little is known about its inner workings. Several attempts have been made to demystify certain aspects of BERT (Rogers et al., 2020), often leading to contradicting conclusions. For instance, Clark et al. (2019) argue that attention measures the importance of a particular word when computing the next level representation for this word. However, Kovaleva et al. (2019) showed that most attention heads contain trivial linguistic information and follow a vertical pattern (attention to [cls], [sep], and punctuation tokens), which could be related to under-utilization or over-parameterization issues. Other studies attempted to link specific BERT heads with linguistically interpretable functions (Htut et al., 2019; Clark et al., 2019; Kovaleva et al., 2019; Voita et al., 2019; Hoover et al., 2020; Lin et al., 2019), agreeing that no single head densely encodes enough relevant information but instead different linguistic features are learnt by different attention heads. We hypothesize that the aforementioned largely contributes to the lack of attention-based explainability of BERT. Another open topic is how the knowledge is distributed across BERT layers. Most studies agree that syntactic knowledge is gathered in the middle layers (Hewitt and Manning, 2019; Goldberg, 2019; Jawahar et al., 2019), while the final layers are more task-specific. Most importantly, it seems that any semantic knowledge is spread across the model, explaining why non-trivial tasks are better solved at the higher layers (Tenney et al., 2019).

Driven by the above discussion, we propose a novel fine-tuning approach where different parts of BERT are guided to directly solve increasingly challenging classification tasks following an underlying label hierarchy. Specifically, we focus on Large Scale Multilabel Text Classification (LMTC) where documents are assigned with one or more labels from a large predefined set. The labels are organized in a hierarchy from general to specific concepts. Our approach attempts to tie specific BERT layers with specific hierarchy levels. In effect, each of these layers is responsible for predicting the labels of the corresponding level. We experiment with two LMTC datasets (EURLEX57K, MIMIC-III) and several variations of structured BERT training. Our contributions are: (a) We propose a novel structured approach to fine-tune BERT where specific layers are tied to specific hierarchy levels; (b) We show that structured training yields better results than the baseline across all levels of the hierarchy, while also leading to better parameter utilization.
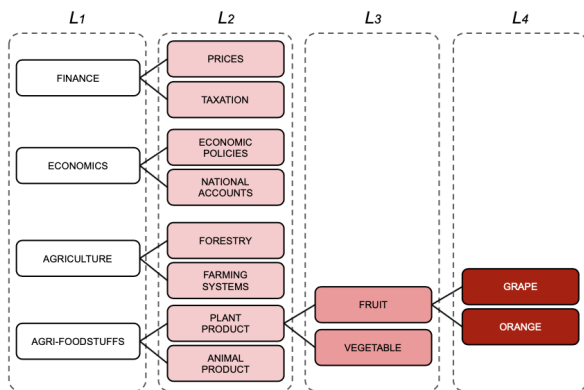
2 Datasets

EURLEX57K (Chalkidis et al., 2019) contains 57k EU legal acts from EURLEX.[1] Each act is approx.

---

[1] http://eur-lex.europa.eu/
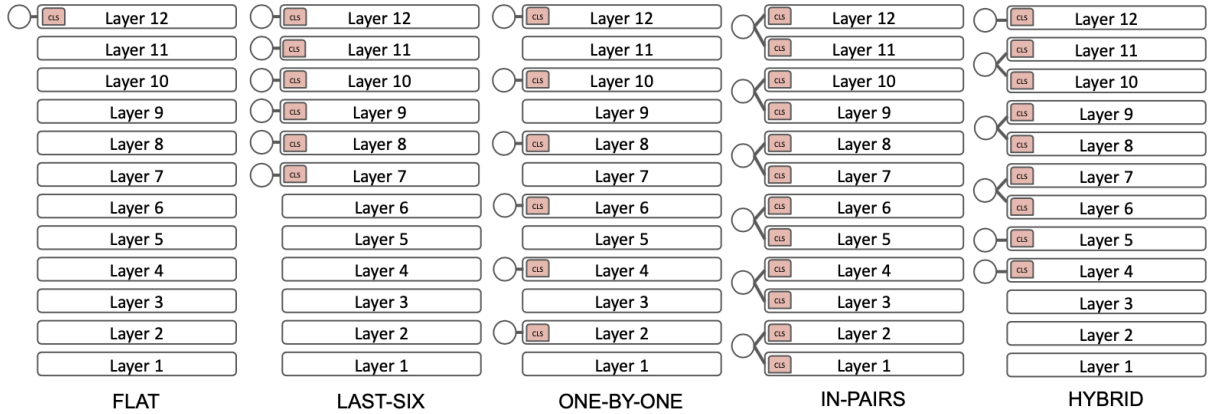
Figure 1: The five variants of BERT-based multi-label classifiers including the flat one and the four structured editions. The circles represent the classification layers attached to [cls] tokens across layers.

Figure 2: Examples from EUROVOC label hierarchy. Layer-wise models consider all labels in the same level (L_i; dashed boxes) of the hierarchy on-par.

700 words long and is annotated with one or more concepts from EUROVOC[2] which contains 7,391 concepts organized in an 8-level hierarchy. We truncate the hierarchy to 6 levels by discarding the last 2 levels which contain 50 rarely used labels.[3]

MIMIC-III (Johnson et al., 2017) contains approx. 52k discharge summaries from US hospitals. Each summary is approx. 1.6k words long and is annotated with one or more ICD-9[4] codes. ICD-9 contains 22,395 codes organized in a 7-level hierarchy. We truncate the hierarchy to 6 levels, discarding the first level which contains only 4 general codes.[3]

Label Augmentation: In both datasets, we make the assumption that if a label l is assigned to a document then all of its ancestors should also be assigned to this document. Hence, we augment labels by annotating a document with all the ancestors of its assigned labels. For instance, in EUROVOC, if a

document is annotated with the label *grape* it will also be annotated with *grape*'s ancestors, i.e., *fruit*, *plant product*, and *agri-foodstuffs* (Figure 2). This assumption is perfectly valid, while also having the added side effect of providing a more accurate test-bed for evaluation. For example, if a classifier mistakenly annotated the document with *citrus fruit*, a sibling of *grape*, in the non-augmented case it would receive a score of zero. By contrast, in the augmented case, assuming it correctly identified all the ancestors of *citrus fruit* it would receive a much higher score of 0.75 having correctly assigned the three ancestors of *grape* but not the (more specialized) label itself. Thus, we believe the model is evaluated more fairly in the augmented case with respect to the hierarchy. This type of evaluation is also in-line with the literature on hierarchical classification (Kosmopoulos et al., 2015).

3 Structured Learning with BERT

Before we proceed with the description of our methods (Figure 1), we introduce some notation. Given a label hierarchy L of depth d, L_n denotes the set of labels in the nth level of this hierarchy ($n \leqslant d$). Also, $f_i = \sigma(W_i \cdot c_i + b_i)$ is a classification function, where W_i and b_i are trainable parameters, c_i is the [cls] token in the ith BERT layer,[5] and σ is the sigmoid activation function. Note that the sizes of W_i and b depend on the number of labels that f_i is responsible for predicting, i.e., if f_i predicts the labels of L_n, $W_i \in \mathbb{R}^{|L_n| \times 768}$ and $b_i \in \mathbb{R}^{|L_n| \times 1}$.

FLAT: This is a simple baseline which uses f_{12} to predict all labels in the hierarchy in a flat manner. In effect, $W_{12} \in \mathbb{R}^{|L| \times 768}$ and $b_i \in \mathbb{R}^{|L| \times 1}$. Note

---

[2]http://eurovoc.europa.eu/

[3]For more details on data manipulation see Appendix A.

[4]www.who.int/classifications/icd/en/

[5]We use BERT-BASE (12 layers, 768 units, 12 heads).

Labels Depth	1	2	3	4	5	6	Micro	Macro
				EURLEX- BERT-BASE				
#Labels	21	127	567	3,861	2,284	481	7,341	7,341
FLAT	90.3 ± 0.2	83.9 ± 0.3	81.0 ± 0.5	74.8 ± 1.0	74.5 ± 1.2	79.9 ± 1.4	80.6 ± 0.6	80.7 ± 0.6
LAST-SIX	$\mathbf{90.7 \pm 0.1}$	$\mathbf{84.6 \pm 0.0}$	$\mathbf{81.9 \pm 0.2}$	$\mathbf{76.8 \pm 0.2}$	$\mathbf{77.2 \pm 0.5}$	$\mathbf{82.2 \pm 1.1}$	$\mathbf{81.7 \pm 0.1}$	$\mathbf{82.2 \pm 0.1}$
ONE-BY-ONE	90.0 ± 0.0	84.3 ± 0.1	81.7 ± 0.2	76.2 ± 0.2	76.7 ± 0.5	81.6 ± 0.2	81.3 ± 0.1	81.7 ± 0.0
IN-PAIRS	89.9 ± 0.2	84.3 ± 0.2	81.7 ± 0.2	76.7 ± 0.3	$\mathbf{77.2 \pm 0.6}$	81.7 ± 0.4	81.4 ± 0.1	81.9 ± 0.4
HYBRID	90.5 ± 0.2	84.3 ± 0.1	81.7 ± 0.2	76.6 ± 0.4	76.6 ± 0.7	81.8 ± 0.7	81.5 ± 0.2	81.9 ± 0.0
				MIMIC-III- SCIBERT				
#Labels	79	589	3,982	9,640	7,234	867	22,391	22,391
FLAT	75.5 ± 0.0	66.5 ± 0.2	57.7 ± 0.4	50.8 ± 0.4	43.2 ± 0.8	38.6 ± 3.3	60.1 ± 0.4	55.4 ± 0.9
LAST-SIX	$\mathbf{76.8 \pm 0.2}$	$\mathbf{67.3 \pm 0.1}$	$\mathbf{58.5 \pm 0.1}$	51.2 ± 0.0	$\mathbf{43.8 \pm 0.3}$	40.9 ± 0.1	$\mathbf{60.4 \pm 0.0}$	$\mathbf{56.4 \pm 0.1}$
ONE-BY-ONE	75.7 ± 0.0	66.7 ± 0.1	57.9 ± 0.1	50.6 ± 0.1	43.4 ± 0.3	$\mathbf{41.8 \pm 0.6}$	59.8 ± 0.1	56.0 ± 0.2
IN-PAIRS	75.6 ± 0.1	66.5 ± 0.1	58.1 ± 0.1	50.9 ± 0.1	43.6 ± 0.3	40.9 ± 0.8	59.9 ± 0.1	55.9 ± 0.2
HYBRID	76.4 ± 0.1	67.0 ± 0.1	$\mathbf{58.5 \pm 0.1}$	$\mathbf{51.3 \pm 0.0}$	$\mathbf{43.8 \pm 0.2}$	40.0 ± 0.4	60.3 ± 0.0	56.2 ± 0.1

Table 1: R-Precision $\pm$ std for all variants of BERT-BASE on EURLEX and MIMIC-III test data across hierarchy depths.

that this model achieves state-of-the-art results on EURLEX57K but not on MIMIC-III (Chalkidis et al., 2020). However, our results are not directly comparable to Chalkidis et al. (2020) because our methods operate on augmented label sets.

LAST-SIX: This method uses the classifiers f_7 through f_{12} to predict the labels in L_1 through L_6, respectively. Our intuition is that the layers 1-6 will retain and enhance their pre-trained functionality, i.e., syntactic knowledge, contextualized representations, while layers 7-12 will leverage this knowledge to better solve their individual tasks. We also expect that the model will show higher parameter utilization for the layers 7-12 since they are forced to solve gradually more refined classification tasks.

ONE-BY-ONE: This method utilizes the full depth of BERT in a *"skip one, use one"* fashion, i.e., it uses classifiers f_i, $i \in \{2, 4, \ldots, 12\}$. In effect, the odd layers $(1, 3, \ldots, 11)$ are updated only indirectly, through the classification tasks of the even layers. We expect that the odd layers will learn rich latent representations to facilitate the classifiers of the even layers. Spreading the classification tasks across the whole depth of the model will potentially lead to better parameter utilization. On the other hand, it could harm the model's pre-trained functionality and hence its performance.

IN-PAIRS: This method also exploits the full depth of BERT, but now the layers are grouped in 6 pairs, $p_n \in \{(1, 2), (3, 4), \ldots, (11, 12)\}$. The classifier responsible for the labels of L_n operates on the concatenated [cls] tokens of the corresponding pair, e.g., $f_1 = \sigma(W_1 \cdot [c_1; c_2] + b_1)$ is trained on the labels of L_1. We expect IN-PAIRS to have better pa-

rameter utilization than ONE-BY-ONE, although the risk of hindering performance is now even higher.

HYBRID: Similarly to LAST-SIX, this method skips some of the lower BERT layers (3 instead of 6). Also, it ties L_1, L_2, and L_6, which are the hierarchy levels with the fewest labels to layers 4, 5, and 12, respectively. Finally, similarly to IN-PAIRS the remaining BERT layers are grouped in pairs and are tied to the rest of the hierarchy levels. We expect the first three layers to retain and enhance their pre-trained functionality, while the hierarchy levels with a large number of labels will benefit from the additional parameters at their disposal.

4 Experiments

We report R-Precision (Manning et al., 2009) at each hierarchy level as well as micro (flat) and macro averages across all levels.[6] Table 1 shows the results in both datasets. In EURLEX57K our structured methods always outperform the baseline mostly by a large margin. LAST-SIX achieves the best overall results and is superior than the other structured methods in all hierarchy levels indicating that allowing the lower layers to retain and enhance their pre-trained functionality is crucial. Similar observations can be made for MIMIC-III, but in this case the importance of not damaging BERT's pre-trained functionality is even higher, as evident by the only minor improvements ONE-BY-ONE and IN-PAIRS have compared to FLAT.[7]

---

[6]See Appendices B and C for a detailed description on experimental setup and a discussion on LMTC evaluation.

[7]This is probably due to the additional difficulties of the clinical domain. See Appendix D for a discussion.
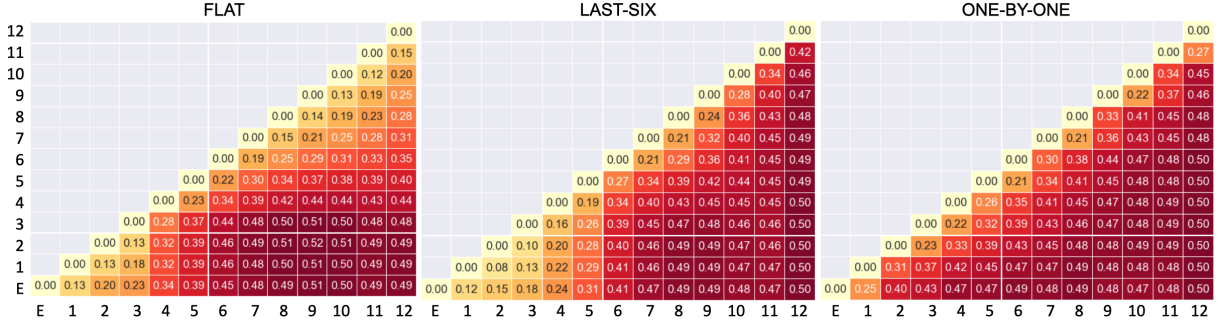
FLAT · LAST-SIX · ONE-BY-ONE

Figure 3: Angular distance between `[cls]` representations across layers.

FLAT

	E	1	2	3	4	5	6	7	8	9	10	11	12
12													0.00
11												0.00	0.15
10											0.00	0.12	0.20
9										0.00	0.13	0.19	0.25
8									0.00	0.14	0.19	0.23	0.28
7								0.00	0.15	0.21	0.25	0.28	0.31
6							0.00	0.19	0.25	0.29	0.31	0.33	0.35
5						0.00	0.22	0.30	0.34	0.37	0.38	0.39	0.40
4					0.00	0.23	0.34	0.39	0.42	0.44	0.44	0.43	0.44
3				0.00	0.28	0.37	0.44	0.48	0.50	0.51	0.50	0.48	0.48
2			0.00	0.13	0.32	0.39	0.46	0.49	0.51	0.52	0.51	0.49	0.49
1		0.00	0.13	0.18	0.32	0.39	0.46	0.48	0.50	0.51	0.50	0.49	0.49
E	0.00	0.13	0.20	0.23	0.34	0.39	0.45	0.48	0.49	0.51	0.50	0.49	0.49

LAST-SIX

	E	1	2	3	4	5	6	7	8	9	10	11	12
12													0.00
11												0.00	0.42
10											0.00	0.34	0.46
9										0.00	0.28	0.40	0.47
8									0.00	0.24	0.38	0.45	0.48
7								0.00	0.21	0.32	0.40	0.45	0.49
6							0.00	0.21	0.29	0.36	0.41	0.45	0.49
5						0.00	0.27	0.34	0.39	0.42	0.44	0.45	0.49
4					0.00	0.19	0.34	0.40	0.43	0.45	0.45	0.46	0.50
3				0.00	0.16	0.26	0.39	0.45	0.47	0.48	0.46	0.46	0.50
2			0.00	0.10	0.20	0.28	0.40	0.46	0.49	0.49	0.47	0.46	0.50
1		0.00	0.08	0.13	0.22	0.29	0.41	0.46	0.49	0.49	0.47	0.47	0.50
E	0.00	0.12	0.15	0.18	0.24	0.31	0.41	0.47	0.49	0.48	0.47	0.47	0.50

ONE-BY-ONE

	E	1	2	3	4	5	6	7	8	9	10	11	12
12													0.00
11												0.00	0.27
10											0.00	0.34	0.45
9										0.00	0.22	0.37	0.46
8									0.00	0.33	0.41	0.45	0.48
7								0.00	0.21	0.36	0.43	0.45	0.48
6							0.00	0.30	0.38	0.44	0.47	0.48	0.50
5						0.00	0.21	0.34	0.41	0.45	0.48	0.48	0.50
4					0.00	0.26	0.35	0.41	0.45	0.48	0.48	0.49	0.50
3				0.00	0.22	0.32	0.39	0.43	0.46	0.47	0.48	0.49	0.50
2			0.00	0.23	0.33	0.39	0.43	0.45	0.48	0.48	0.49	0.49	0.50
1		0.00	0.31	0.37	0.42	0.45	0.47	0.47	0.49	0.48	0.48	0.48	0.50
E	0.00	0.25	0.40	0.43	0.47	0.47	0.48	0.49	0.48	0.48	0.47	0.48	0.50

FLAT · LAST-SIX · ONE-BY-ONE

Figure 4: KL-Divergence between attention distributions across layers.

FLAT

	1	2	3	4	5	6	7	8	9	10	11	12
12												0.0
11											0.0	0.3
10										0.0	0.2	0.4
9									0.0	0.3	0.2	0.6
8								0.0	0.2	0.3	0.3	0.7
7							0.0	0.3	0.4	0.5	0.5	0.8
6						0.0	0.5	0.5	0.5	0.8	0.6	0.9
5					0.0	0.3	0.8	0.7	0.6	0.9	0.6	0.9
4				0.0	0.3	0.6	0.9	1.0	0.9	1.0	0.9	1.1
3			0.0	0.4	1.1	1.4	1.3	1.4	1.5	1.5	1.5	1.7
2		0.0	0.3	0.5	1.5	1.7	1.6	1.9	2.3	2.0	2.2	2.2
1	0.0	0.5	1.1	1.0	1.7	1.9	1.7	2.2	2.7	2.4	2.7	2.6

LAST-SIX

	1	2	3	4	5	6	7	8	9	10	11	12
12												0.0
11											0.0	1.3
10										0.0	0.6	0.7
9									0.0	0.9	1.4	1.7
8								0.0	0.7	1.7	1.7	3.9
7							0.0	0.3	0.8	1.6	1.6	3.6
6						0.0	0.5	0.5	0.9	1.8	1.8	3.9
5					0.0	0.5	0.5	0.5	0.6	1.4	1.7	2.7
4				0.0	0.5	0.7	0.7	0.6	0.8	1.2	1.1	
3			0.0	0.4	1.4	1.5	1.6	1.3	0.9	1.3	0.8	
2		0.0	0.2	0.6	1.7	1.8	1.8	2.0	1.6	1.0	1.3	0.6
1	0.0	0.4	0.6	0.8	1.9	1.9	1.9	2.1	1.7	1.1	1.4	0.5

ONE-BY-ONE

	1	2	3	4	5	6	7	8	9	10	11	12
12												0.0
11											0.0	0.3
10										0.0	0.3	0.5
9									0.0	0.4	0.5	0.5
8								0.0	0.7	0.5	0.7	0.5
7							0.0	0.6	1.0	0.9	0.8	0.7
6						0.0	1.7	1.8	4.0	3.1	3.5	2.5
5					0.0	0.6	0.8	0.8	1.8	1.5	1.7	1.3
4				0.0	1.1	1.5	1.3	1.9	3.1	2.7	2.7	2.1
3			0.0	1.0	1.7	2.3	1.5	2.3	3.4	3.0	2.8	2.4
2		0.0	0.8	0.9	1.9	2.7	1.7	2.6	3.5	3.2	2.9	2.4
1	0.0	0.9	1.2	1.2	1.9	2.8	1.3	2.2	2.7	2.5	2.1	1.8

Layer	1		2		3		4		5		6		7		8		9		10		11		12	
FLAT	4.7	1.3	3.9	0.7	2.5	0.9	3.6	1.7	3.3	2.6	3.2	3.2	3.4	3.1	2.7	2.6	2.1	2.1	2.6	1.8	2.2	1.2	2.4	0.6
LAST-SIX	4.6	**1.5**	**4.2**	0.6	**4.1**	0.8	**3.9**	1.1	2.7	**2.7**	2.9	**3.6**	2.9	3.0	2.6	**2.9**	**3.0**	**4.2**	**4.5**	**3.9**	**4.3**	**2.3**	**5.2**	0.6
ONE-BY-ONE	4.5	**1.9**	**4.1**	**2.4**	**4.0**	**4.7**	**4.0**	**3.8**	**3.7**	**4.2**	2.8	2.9	**4.6**	2.7	**4.0**	**3.0**	**3.9**	1.7	**3.9**	1.8	**4.4**	**1.4**	**4.5**	0.2
IN-PAIRS	4.4	**1.9**	3.9	**3.0**	3.9	**4.6**	**4.1**	**4.2**	3.2	**3.6**	**3.5**	**3.5**	**4.1**	2.5	**3.5**	**2.8**	**3.7**	2.0	**3.7**	1.4	**4.2**	1.1	**5.1**	0.2
HYBRID	4.6	**2.0**	**4.5**	0.9	**4.2**	**1.7**	**4.2**	**2.1**	**3.8**	**3.4**	3.1	**3.7**	**4.1**	**3.8**	**3.8**	**3.6**	**4.1**	**2.8**	**4.3**	**2.5**	**4.1**	1.1	**4.7**	0.3

Table 2: Entropy (left column) and KL-Divergence (right column) of attention distributions per layer.

5 Discussion on model utilization

Comparing `[cls]` representations: Figure 3 shows the average angular distances between the `[cls]` representations of each layer on development data of EURLEX57K.[8] The angular distance is calculated on unit (L_2 normalized) vectors, takes values in $[0, 1]$, and a distance of 0.5 indicates an angle of $90°$. We observe that ONE-BY-ONE leads to larger angles between the representations than LAST-SIX which in turn yields larger angles than FLAT. In effect, ONE-BY-ONE and to a lesser extent LAST-SIX lead to a better parameter utilization than FLAT. To better support this claim we provide a geometric interpretation. We first L_2 normalize all `[cls]` representations. Each normalized representation can be interpreted as a vector having its initial point at the origin and its terminal point at the surface of a 768-dimensional hyper-sphere (centered at the origin). The larger the angle between two `[cls]` vectors the further apart they are on the hyper-sphere's surface. Effectively, `[cls]` vectors with large angles between them cover a larger sub-area of the hyper-sphere's surface indicating that the vector space is utilized to a higher extent which directly implies better parameter utilization.

Comparing attention distributions: Figure 4 shows the KL-Divergence of the average (across heads) attention for all layers on the development data of EURLEX57K.[8] A high KL-Divergence indicates that two layers attend to different sub-word units. Moreover, Table 2 reports the entropy (left column per layer) of the average (across heads) attention distribution per layer. A high entropy indicates that a layer attends to more sub-word units. Table 2 also reports the average KL-Divergence (right column per layer) between the attention distributions of each possible pair of heads in a layer. A high KL-Divergence indicates that each head attends to different sub-word units. A first observation is that FLAT attends to almost the same sub-word units across layers (small entropy differences and KL-Divergence across layers). Interestingly,

---

[8]For brevity, we only show the heatmaps of three methods. We include the missing ones in Appendix E.

the different attention heads focus on different sub-word units only in the middle layers (5-8). On the other hand, all the structured methods show better utilization of the attention mechanism, having higher entropy and KL-Divergence both across heads (Table 2) and across layers (Figure 4).

6 Related Work

Our approach is similar to Wehrmann et al. (2018) but they experiment with fully connected networks, which are not well suited for text classification, contrary to stacked transformers (Vaswani et al., 2017; Devlin et al., 2019). Similarly, Yan et al. (2015) used Convolutional Neural Networks, albeit with shallow hierarchies (2 levels). Although our approach leverages the label hierarchy it should not be confused with hierarchical classification methods (Silla and Freitas, 2011), which typically employ one classifier per node and cannot scale-up to large hierarchies when considering neural classifiers. A notable exception is the work of You et al. (2019) who employed one bidirectional LSTM with label-wise attention (You et al., 2018) per hierarchy node. However, for their method to scale-up, they use probabilistic label trees (Khandagale et al., 2019) to organize the labels in their own shallow hierarchy which does not follow the abstraction level of the original hierarchy. To the best of our knowledge we are the first to apply this approach to pre-trained language models.

7 Conclusions and Further Work

We proposed a novel guided approach to fine-tune BERT, where specific layers are tied to specific hierarchy levels. Experimenting with two LMTC datasets, we showed that structured training not only yields better results than a flat baseline, but also leads to better parameter utilization. In the future we will try to further increase the parameter utilization by guiding BERT's attention heads to explicitly focus on specific hierarchy parts. We also plan to improve the explainability of our methods with respect to the utilization of their parameters.

References

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. SciB-ERT: A pretrained language model for scientific text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Lan-guage Processing (EMNLP-IJCNLP)*, pages 3606–3611, Hong Kong, China.

Ilias Chalkidis, Emmanouil Fergadiotis, Prodromos Malakasiotis, and Ion Androutsopoulos. 2019. Large-Scale Multi-Label Text Classification on EU Legislation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6314–6322, Florence, Italy.

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

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. 2019. What does BERT look at? an analysis of BERT's attention. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 276–286, Florence, Italy. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, abs/1810.04805.

Yoav Goldberg. 2019. Assessing bert's syntactic abilities.

John Hewitt and Christopher D. Manning. 2019. A structural probe for finding syntax in word representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4129–4138, Minneapolis, Minnesota. Association for Computational Linguistics.

Benjamin Hoover, Hendrik Strobelt, and Sebastian Gehrmann. 2020. exBERT: A Visual Analysis Tool to Explore Learned Representations in Transformer Models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 187–196, Online. Association for Computational Linguistics.

Phu Mon Htut, Jason Phang, Shikha Bordia, and Samuel R. Bowman. 2019. Do attention heads in bert track syntactic dependencies?

Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. 2019. What does BERT learn about the structure of language? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3651–3657, Florence, Italy. Association for Computational Linguistics.

Alistair EW Johnson, David J. Stone, Leo A. Celi, and Tom J. Pollard. 2017. MIMIC-III, a freely accessible critical care database. *Nature*.

Sujay Khandagale, Han Xiao, and Rohit Babbar. 2019. Bonsai - Diverse and Shallow Trees for Extreme Multi-label Classification. *CoRR*, abs/1904.08249.

Diederik P. Kingma and Jim Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of the 5th International Conference on Learning Representations*.

Aris Kosmopoulos, Ioannis Partalas, Eric Gaussier, Georgios Paliouras, and Ion Androutsopoulos. 2015. Evaluation measures for hierarchical classification: a unified view and novel approaches. *Data Mining and Knowledge Discovery*, 29(3):820–865.

Olga Kovaleva, Alexey Romanov, Anna Rogers, and Anna Rumshisky. 2019. Revealing the dark secrets of BERT. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4365–4374, Hong Kong, China. Association for Computational Linguistics.

Yongjie Lin, Yi Chern Tan, and Robert Frank. 2019. Open sesame: Getting inside BERT's linguistic knowledge. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 241–253, Florence, Italy. Association for Computational Linguistics.

Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. 2009. *Introduction to Information Retrieval*. Cambridge University Press.

Anthony Rios and Ramakanth Kavuluru. 2018. Few-Shot and Zero-Shot Multi-Label Learning for Structured Label Spaces. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3132–3142. Association for Computational Linguistics.

Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2020. A primer in bertology: What we know about how bert works. *arXiv preprint arXiv:2002.12327*.

Carlos N Silla and Alex A Freitas. 2011. A survey of hierarchical classification across different application domains. *Data Mining and Knowledge Discovery*, 22(1-2):31–72.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019. BERT rediscovers the classical NLP pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4593–4601, Florence, Italy. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention Is All You Need. In *31th Annual Conference on Neural Information Processing Systems*, Long Beach, CA, USA.

Elena Voita, David Talbot, Fedor Moiseev, Rico Sennrich, and Ivan Titov. 2019. Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5797–5808, Florence, Italy. Association for Computational Linguistics.

Jonatas Wehrmann, Ricardo Cerri, and Rodrigo Barros. 2018. Hierarchical multi-label classification networks. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 5075–5084, Stockholmsmässan, Stockholm Sweden. PMLR.

Zhicheng Yan, Hao Zhang, Robinson Piramuthu, Vignesh Jagadeesh, Dennis DeCoste, Wei Di, and Yizhou Yu. 2015. Hd-cnn: Hierarchical deep convolutional neural networks for large scale visual recognition. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 2740–2748.

Ronghui You, Suyang Dai, Zihan Zhang, Hiroshi Mamitsuka, and Shanfeng Zhu. 2018. Attentionxml: Extreme multi-label text classification with multi-label attention based recurrent neural networks. *arXiv preprint arXiv:1811.01727*.

Ronghui You, Zihan Zhang, Ziye Wang, Suyang Dai, Hiroshi Mamitsuka, and Shanfeng Zhu. 2019. Attentionxml: Label tree-based attention-aware deep model for high-performance extreme multi-label text classification. In *Advances in Neural Information Processing Systems 32*, pages 5812–5822. Curran Associates, Inc.

A   Data manipulation

Hierarchy truncation: In order to directly apply all our methods, we truncate both hierarchies and reduce their depth to six. We believe this truncation is justified since in EUROVOC the last two layers contain a very small number of labels, which are rarely, if at all, assigned and in ICD-9 the first layer also contains a very small number of labels which are very general and can be trivially classified (Table 3). In both cases it seems that only minimal information is lost which would have small practical use in the classification tasks.

Document Truncation: Documents in both datasets are often above the 512 token limit of BERT. To reduce document size, we perform a number of pre-processing normalizations, including removal of numeric tokens, punctuation and stop-words.[9] In EURLEX documents have been tokenized using SpaCy's default tokenizer,[10] while in

---

[9] Similar procedures are very common in classification, thus we believe they do not harm text semantics.

[10] https://spacy.io

Depth	EUROVOC	ICD-9
1	21	4*
2	127	79
3	568	589
4	4,545	3,982
5	2,335	9,640
6	497	7,234
7	79*	867
8	6*	-
Overall	8,178 / 8,093	22,395 / 22,391

Table 3: Label distribution across EUROVOC and ICD-9 hierarchy levels. Concepts (labels) are arranged from more abstract (level 1-2) to more specialized ones (levels 6-8). Labels with an asterisk are truncated in our experiments.

MIMIC-III, we use regular expressions tailored for the biomedical domain. While document length is severely reduced post normalization, if a document still has a larger number of tokens, i.e. more than 512, we use the first 512 tokens and ignore the rest.

B Experimental Setup

All our methods build on BERT-BASE and are implemented in Tensorflow 2. For EURLEX we use the original BERT-BASE (Devlin et al., 2019), while for MIMIC-III we use SCIBERT (Beltagy et al., 2019), which has the same architecture (12 layers, 768 hidden units, 12 attention heads), and better suits biomedical documents.[11] Our models are tuned by grid searching three learning rates ($2e\text{-}5, 3e\text{-}5, 5e\text{-}5$) and two drop-out rates ($0, 0.1$). We use the Adam optimizer (Kingma and Ba, 2015) with early stopping on validation loss. In preliminary experiments, we found that weighting individual losses with respect to the number of labels in each level is crucial. We therefore weigh each loss by the percentage of labels at the corresponding level, i.e., $w_n = \frac{|L_n|}{|L|}$, where $|L_n|$ is the number of labels in the nth level of the hierarchy and $|L|$ is the total number of labels across all levels, e.g., in EURLEX57K, $w_1 = \frac{21}{8093} \approx 0.0026$.

C Evaluation in LMTC

The literature of LMTC (Rios and Kavuluru, 2018; Chalkidis et al., 2019) mostly uses information retrieval evaluation measures. We support the premise that when the number of labels is that large the problem mimics retrieval with each document acting as a query and the model having to score relevant labels higher than the rest. However in our study, it would be really confusing

---

[11]We use the Transformers library of Huggingface (https://github.com/huggingface/transformers).

to report the standard retrieval metrics Recall@R, Precision@K, nDCG@K since we evaluate our classifiers at each hierarchy depth and reasonable values for K have large fluctuations between levels, as the number of labels per level vastly varies (see Table 3). Instead, we prefer R-Precision (Manning et al., 2009), which is the Precision@R where R is the number of gold labels associated with each document. It follows that R-Precision can neither under-estimate (penalize) nor over-estimate the performance of the models (Chalkidis et al., 2019).

D Peculiarities of MIMIC-III dataset

In our experiments we observe a hindered performance in MIMIC-III, which can be attributed to a number of characteristics of the dataset. Firstly, documents contain a lot of non-trivial biomedical terminology which naturally makes the classification task more difficult. Further, discharge summaries describe a patient's condition during their hospitalization and therefore proper label annotations change throughout the document as the patient's diagnosis changes or as they exhibit new symptoms, e.g., "*the patient was admitted to the hospital with <u>no heart issues</u>, [. . .] the patient had <u>a heart failure and died</u>.*". Both the in-domain language and the constant change of events make the dataset more challenging than EURLEX57K, where documents are more organized and well-written also with simpler language.

It therefore seems reasonable that in MIMIC-III allowing lower BERT layers to retain and enhance the preliminary functionality, without explicitly guiding them, is of utmost importance. We would like to highlight that even though we use SCIBERT (Beltagy et al., 2019), which is based on a new scientific vocabulary, we observe that specialized biomedical terms are often over-fragmented in multiple sub-word units, e.g. 'atelectasis' splits into ['ate', '##lect', '##asis']. Thus, the initial layers need to decipher these over-fragmented sub-word units and reconstruct the original word semantics. On the contrary, in EURLEX57K, classifying general concepts in the initial layers, even considering only the sub-word unit embeddings is plausible.

E Discussion on model utilization

We present additional results for the rest of the methods (IN-PAIRS, HYBRID). Figure 5 shows the average angular distances between the [cls] representations of each layer (Figure 5) for all con-
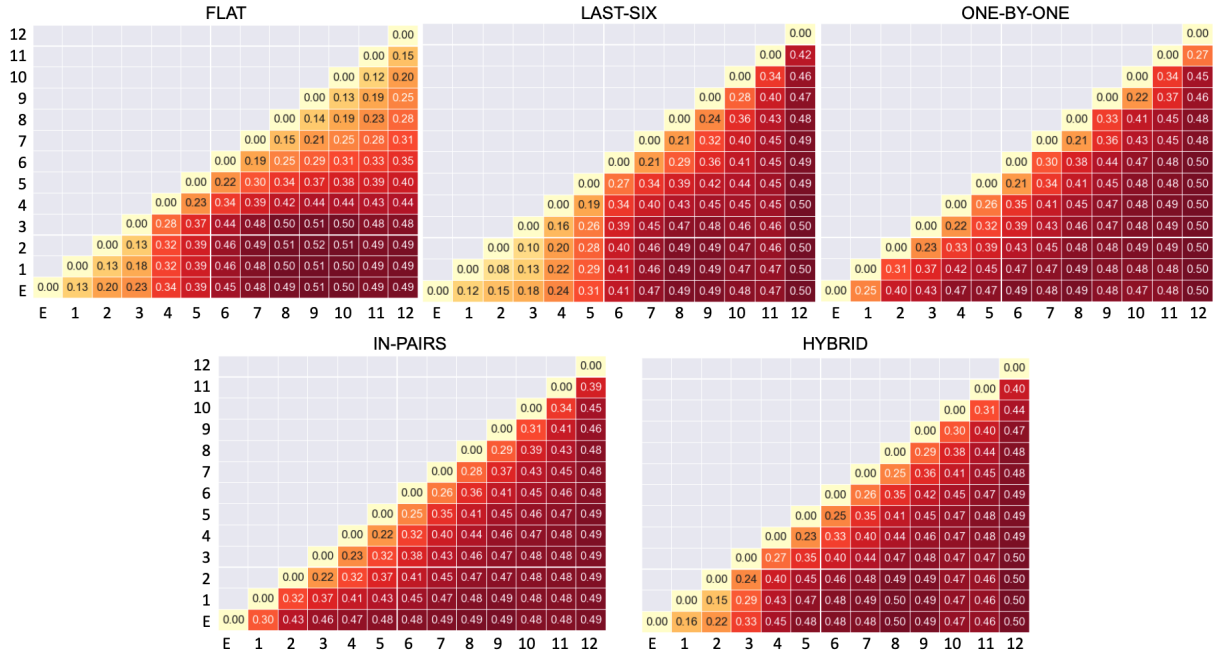
Figure 5: Angular distance between `[cls]` representations across layers in the development dataset of EURLEX57K.
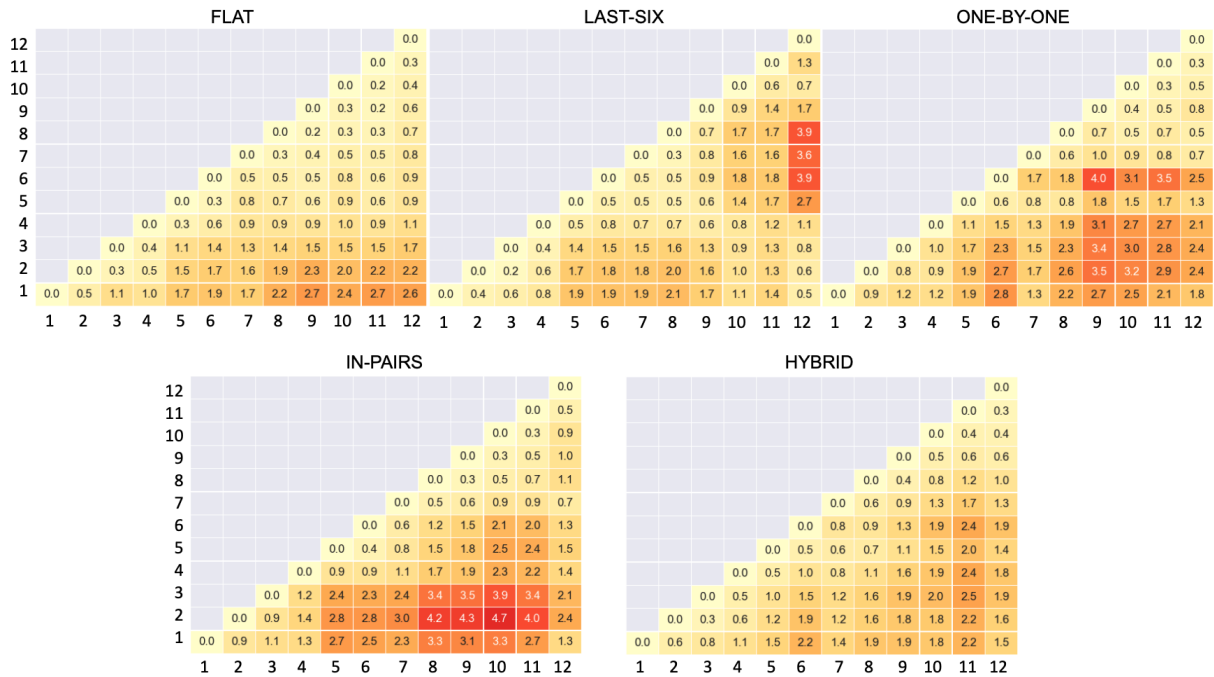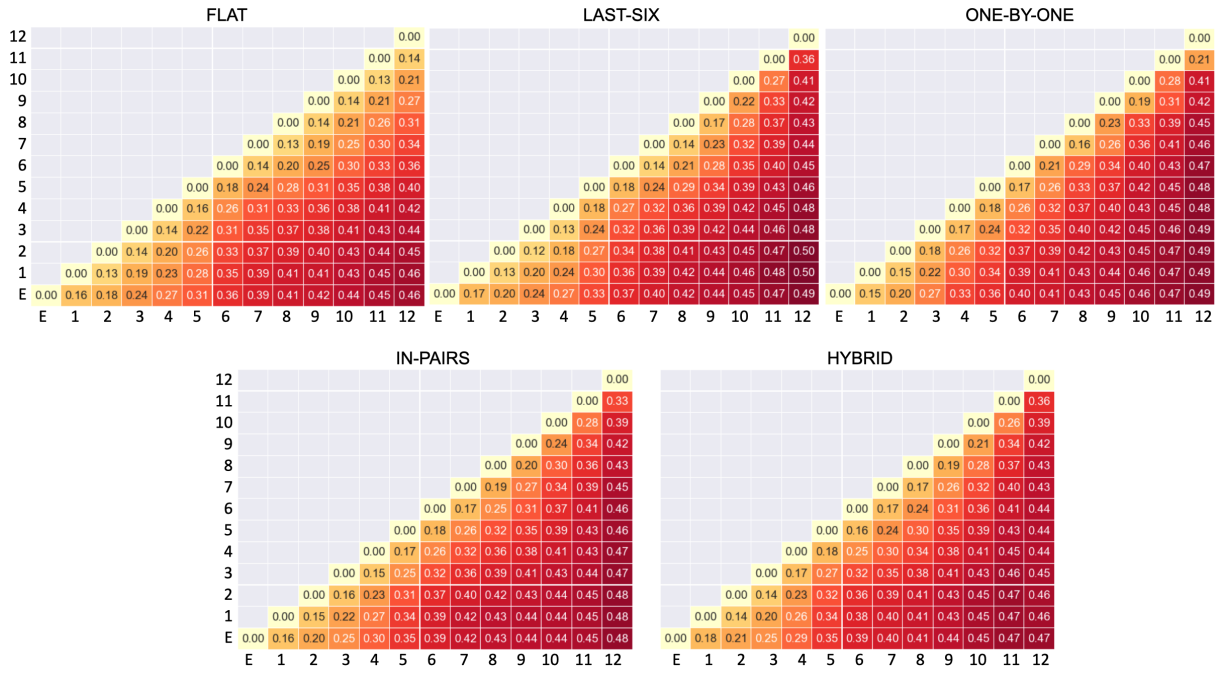

Figure 6: KL-Divergence between attention distributions across layers in the development dataset of EURLEX57K.

sidered methods. We observe that the distances of IN-PAIRS between consecutive `[cls]` representations follow a similar pattern with those of ONE-BY-ONE, with the exception of 0.25+ distances which are more dense in the upper layers for IN-PAIRS. This is reasonable, since in IN-PAIRS all layers directly contribute to the classification tasks. The pattern of HYBRID is very similar to ONE-BY-ONE and IN-PAIRS, except for the first three non-guided

layers in which distances bear close resemblance to those of the corresponding layers in LAST-SIX. Similar observations hold for MIMIC-III (Figure 7). Finally, Figure 6 shows the KL-Divergence of the average (across heads) attention for all layers on the development data. All structured methods show better utilization of the attention mechanism than FLAT, having higher KL-Divergence across layers. Contrary, in MIMIC-III, all structured methods fol-

60

Figure 7: Angular distance between `[cls]` representations across layers in the development dataset of MIMIC-III.
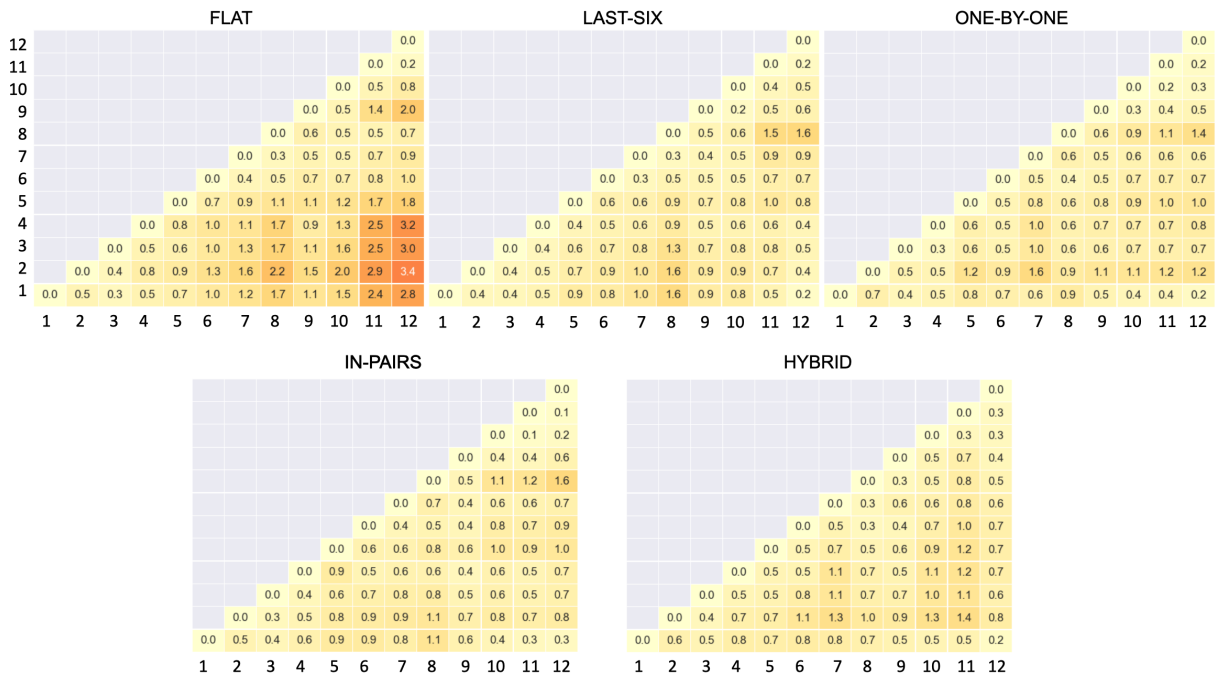
Figure 8: KL-Divergence between attention distributions across layers in the development dataset of MIMIC-III.

low a similar pattern of low KL-Divergence across layers (Figure 8), even lower than the upper layers of FLAT, i.e., the models attend to similar sub-word positions across layers. We aim to further study and explain this behaviour in future work.