Large-Scale Multi-Label Text Classification on EU Legislation

Ilias Chalkidis **Manos Fergadiotis** **Prodromos Malakasiotis**
Ion Androutsopoulos
Department of Informatics, Athens University of Economics and Business, Greece
[ihalk,fergadiotis,rulller,ion]@aueb.gr

Abstract

We consider Large-Scale Multi-Label Text Classification (LMTC) in the legal domain. We release a new dataset of 57k legislative documents from EUR-LEX, annotated with ∼4.3k EUROVOC labels, which is suitable for LMTC, few- and zero-shot learning. Experimenting with several neural classifiers, we show that BIGRUs with label-wise attention perform better than other current state of the art methods. Domain-specific WORD2VEC and context-sensitive ELMO embeddings further improve performance. We also find that considering only particular zones of the documents is sufficient. This allows us to bypass BERT's maximum text length limit and fine-tune BERT, obtaining the best results in all but zero-shot learning cases.

1 Introduction

Large-scale multi-label text classification (LMTC) is the task of assigning to each document all the relevant labels from a large set, typically containing thousands of labels (classes). Applications include building web directories (Partalas et al., 2015), labeling scientific publications with concepts from ontologies (Tsatsaronis et al., 2015), assigning diagnostic and procedure labels to medical records (Mullenbach et al., 2018; Rios and Kavuluru, 2018). We focus on legal text processing, an emerging NLP field with many applications (e.g., legal judgment (Nallapati and Manning, 2008; Aletras et al., 2016), contract element extraction (Chalkidis et al., 2017), obligation extraction (Chalkidis et al., 2018)), but limited publicly available resources.

Our first contribution is a new publicly available legal LMTC dataset, dubbed EURLEX57K, containing 57k English EU legislative documents from the EUR-LEX portal, tagged with ∼4.3k labels (concepts) from the European Vocabulary (EUROVOC).[1] EUROVOC contains approx. 7k labels, but most of them are rarely used, hence they are under-represented (or absent) in EURLEX57K, making the dataset also appropriate for few- and zero-shot learning. EURLEX57K can be viewed as an improved version of the dataset released by Mencia and Fürnkranzand (2007), which has been widely used in LMTC research, but is less than half the size of EURLEX57K (19.6k documents, 4k EUROVOC labels) and more than ten years old.

As a second contribution, we experiment with several neural classifiers on EURLEX57K, including the Label-Wise Attention Network of Mullenbach et al. (2018), called CNN-LWAN here, which was reported to achieve state of the art performance in LMTC on medical records. We show that a simpler BIGRU with self-attention (Xu et al., 2015) outperforms CNN-LWAN by a wide margin on EURLEX57K. However, by replacing the CNN encoder of CNN-LWAN with a BIGRU, we obtain even better results on EURLEX57K. Domain-specific WORD2VEC (Mikolov et al., 2013) and context-sensitive ELMO embeddings (Peters et al., 2018) yield further improvements. We thus establish strong baselines for EURLEX57K.

As a third contribution, we investigate which zones of the documents are more informative on EURLEX57K, showing that considering only the title and recitals of each document leads to almost the same performance as considering the full document. This allows us to bypass BERT's (Devlin et al., 2018) maximum text length limit and fine-tune BERT, obtaining the best results for all but zero-shot learning labels. To our knowledge, this is the first application of BERT to an LMTC task, which provides further evidence of the superiority of pretrained language models with task-specific

[1] See https://eur-lex.europa.eu/ for EUR-LEX, and https://publications.europa.eu/en/web/eu-vocabularies for EUROVOC.

fine-tuning, and establishes an even stronger baseline for EURLEX57K and LMTC in general.

2 Related Work

You et al. (2018) explored RNN-based methods with self-attention on five LMTC datasets that had also been considered by Liu et al. (2017), namely RCV1 (Lewis et al., 2004), Amazon-13K, (McAuley and Leskovec, 2013), Wiki-30K and Wiki-500K (Zubiaga, 2012), as well as the previous EUR-LEX dataset (Mencia and Fürnkranzand, 2007), reporting that attention-based RNNs produced the best results overall (4 out of 5 datasets).

Mullenbach et al. (2018) investigated the use of label-wise attention in LMTC for medical code prediction on the MIMIC-II and MIMIC-III datasets (Johnson et al., 2017). Their best method, Convolutional Attention for Multi-Label Classification, called CNN-LWAN here, employs one attention head per label and was shown to outperform weak baselines, namely logistic regression, plain BIGRUs, CNNs with a single convolution layer.

Rios and Kavuluru (2018) consider few- and zero-shot learning on the MIMIC datasets. They propose Zero-shot Attentive CNN, called ZERO-CNN-LWAN here, a method similar to CNN-LWAN, which also exploits label descriptors. Although ZERO-CNN-LWAN did not outperform CNN-LWAN overall on MIMIC-II and MIMIC-III, it had much improved results in few-shot and zero-shot learning, among other variations of ZERO-CNN-LWAN that exploit the hierarchical relations of the labels with graph convolutions.

We note that the label-wise attention methods of Mullenbach et al. (2018) and Rios and Kavuluru (2018) were not compared to strong generic text classification baselines, such as attention-based RNNs (You et al., 2018) or Hierarchical Attention Network (HAN) (Yang et al., 2016), which we investigate below.

3 The New Dataset

As already noted, EURLEX57K contains 57k legislative documents from EUR-LEX[2] with an average length of 727 words (Table 1).[3] Each document contains four major zones: the *header*, which includes the title and name of the legal body enforcing the legal act; the *recitals*, which are legal background references; the *main body*, usually organized in articles; and the *attachments* (e.g., appendices, annexes).

Subset	Documents (D)	Words/D	Labels/D
Train	45,000	729	5
Dev.	6,000	714	5
Test	6,000	725	5
Total	57,000	727	5

Table 1: Statistics of the EUR-LEX dataset.

Some of the LMTC methods we consider need to be fed with documents split into smaller units. These are often sentences, but in our experiments they are *sections*, thus we preprocessed the raw text, respectively. We treat the header, the recitals zone, each article of the main body, and the attachments as separate sections.

All the documents of the dataset have been annotated by the Publications Office of EU[4] with multiple concepts from EUROVOC. While EUROVOC includes approx. 7k concepts (labels), only 4,271 (59.31%) are present in EURLEX57K, from which only 2,049 (47.97%) have been assigned to more than 10 documents. Similar distributions were reported by Rios and Kavuluru (2018) for the MIMIC datasets. We split EURLEX57K into training (45k documents), development (6k), and test subsets (6k). We also divide the 4,271 labels into *frequent* (746 labels), *few-shot* (3,362), and *zero-shot* (163), depending on whether they were assigned to more than 50, fewer than 50 but at least one, or no training documents, respectively.

4 Methods

Exact Match, Logistic Regression: A first naive baseline, Exact Match, assigns only labels whose descriptors can be found verbatim in the document. A second one uses Logistic Regression with feature vectors containing TF-IDF scores of n-grams ($n = 1, 2, \ldots, 5$).

BIGRU-ATT: The first neural method is a BIGRU with self-attention (Xu et al., 2015). Each document is represented as the sequence of its word embeddings, which go through a stack of BIGRUs (Figure 1a). A document embedding (h) is computed as the sum of the resulting context-aware embeddings ($h = \sum_i a_i h_i$), weighted by the self-attention scores (a_i), and goes through a dense

[2] Our dataset is available at http://nlp.cs.aueb.gr/software_and_datasets/EURLEX57K, with permission of reuse under European Union©, https://eur-lex.europa.eu, 1998–2019.

[3] See Appendix A for more statistics.

[4] See https://publications.europa.eu/en.

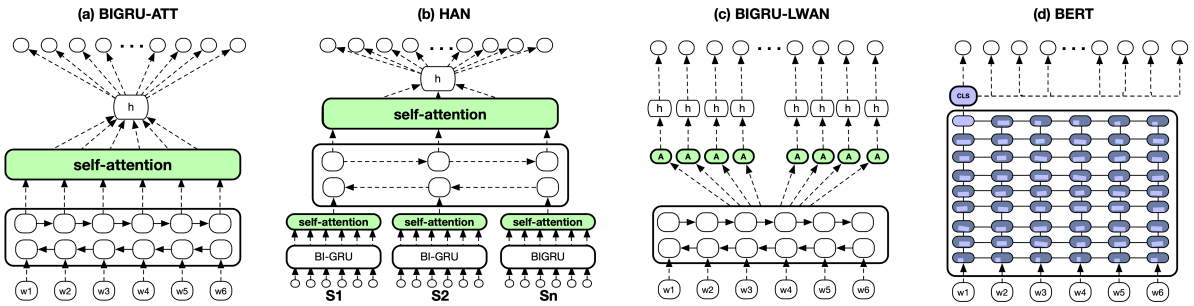

Figure 1: Illustration of (a) BIGRU-ATT, (b) HAN, (c) BIGRU-LWAN, and (d) BERT.

layer of $L = 4,271$ output units with sigmoids, producing L probabilities, one per label.

HAN: The Hierarchical Attention Network (Yang et al., 2016) is a strong baseline for text classification. We use a slightly modified version, where a BIGRU with self-attention reads the words of each section, as in BIGRU-ATT but separately per section, producing section embeddings. A second-level BIGRU with self-attention reads the section embeddings, producing a single document embedding (h) that goes through a similar output layer as in BIGRU-ATT (Figure 1b).

CNN-LWAN, BIGRU-LWAN: In the original Label-Wise Attention Network (LWAN) of Mullenbach et al. (2018), called CNN-LWAN here, the word embeddings of each document are first converted to a sequence of vectors h_i by a CNN encoder. A modified version of CNN-LWAN that we developed, called BIGRU-LWAN, replaces the CNN encoder with a BIGRU (Figure 1c), which converts the word embeddings into context-sensitive embeddings h_i, much as in BIGRU-ATT. Unlike BIGRU-ATT, however, both CNN-LWAN and BIGRU-LWAN use L independent attention heads, one per label, generating L document embeddings ($h^{(l)} = \sum_i a_{l,i} h_i$, $l = 1, \ldots, L$) from the sequence of vectors h_i produced by the CNN or BIGRU encoder, respectively. Each document embedding ($h^{(l)}$) is specialized to predict the corresponding label and goes through a separate dense layer (L dense layers in total) with a sigmoid, to produce the probability of the corresponding label.

ZERO-CNN-LWAN, ZERO-BIGRU-LWAN: Rios and Kavuluru (2018) designed a model similar to CNN-LWAN, called ZACNN in their work and ZERO-CNN-LWAN here, to deal with rare labels. In ZERO-CNN-LWAN, the attention scores ($a_{l,i}$) and the label probabilities are produced by comparing the h_i vectors that the CNN encoder produces and the label-specific document embeddings ($h^{(l)}$), respectively, to label embeddings. Each label embedding is the centroid of the pretrained word embeddings of the label's descriptor; consult Rios and Kavuluru (2018) for further details. By contrast, CNN-LWAN and BIGRU-LWAN do not consider the descriptors of the labels. We also experiment with a variant of ZERO-CNN-LWAN that we developed, dubbed ZERO-BIGRU-LWAN, where the CNN encoder is replaced by a BIGRU.

BERT: BERT (Devlin et al., 2018) is a language model based on Transformers (Vaswani et al., 2017) pretrained on large corpora. For a new target task, a task-specific layer is added on top of BERT. The extra layer is trained jointly with BERT by fine-tuning on task-specific data. We add a dense layer on top of BERT, with sigmoids, that produces a probability per label. Unfortunately, BERT can currently process texts up to 512 word-pieces, which is too small for the documents of EURLEX57K. Hence, BERT can only be applied to truncated versions of our documents (see below).

5 Experiments

Evaluation measures: Common LMTC evaluation measures are precision ($P@K$) and recall ($R@K$) at the top K predicted labels, averaged over test documents, micro-averaged F1 over all labels, and $nDCG@K$ (Manning et al., 2009). However, $P@K$ and $R@K$ unfairly penalize methods when the gold labels of a document are fewer or more than K, respectively. Similar concerns have led to the introduction of R-Precision and $nDCG@K$ in Information Retrieval (Manning et al., 2009), which we believe are also more appropriate for LMTC. Note, however, that R-Precision requires the number of gold labels per document to be known beforehand, which is unrealistic in practical applications. Therefore we propose using R-Precision@K ($RP@K$), where

	All Labels			Frequent		Few		Zero	
	$RP@5$	$nDCG@5$	Micro-$F1$	$RP@5$	$nDCG@5$	$RP@5$	$nDCG@5$	$RP@5$	$nDCG@5$
Exact Match	0.097	0.099	0.120	0.219	0.201	0.111	0.074	0.194	0.186
Logistic Regression	0.710	0.741	0.539	0.767	0.781	0.508	0.470	0.011	0.011
BIGRU-ATT	0.758	0.789	0.689	0.799	0.813	0.631	0.580	0.040	0.027
HAN	0.746	0.778	0.680	0.789	0.805	0.597	0.544	0.051	0.034
CNN-LWAN	0.716	0.746	0.642	0.761	0.772	0.613	0.557	0.036	0.023
BIGRU-LWAN	**0.766**	**0.796**	**0.698**	**0.805**	**0.819**	**0.662**	**0.618**	0.029	0.019
ZERO-CNN-LWAN	0.684	0.717	0.618	0.730	0.745	0.495	0.454	0.321	0.264
ZERO-BIGRU-LWAN	0.718	0.752	0.652	0.764	0.780	0.561	0.510	**0.438**	**0.345**
BIGRU-LWAN (L2V)	0.775	0.804	0.711	0.815	0.828	0.656	0.612	0.034	0.024
BIGRU-LWAN (L2V) *	0.770	0.796	0.709	0.811	0.825	0.641	0.600	0.047	0.030
BIGRU-LWAN (ELMO) *	0.781	0.811	0.719	0.821	0.835	0.668	0.619	0.044	0.028
BERT-BASE *	**0.796**	**0.823**	**0.732**	**0.835**	**0.846**	**0.686**	**0.636**	0.028	0.023

Table 2: Results on EURLEX57K for all, frequent, few-shot, zero-shot labels. Starred methods use the first 512 document tokens; all other methods use full documents. Unless otherwise stated, GLOVE embeddings are used.

K is a parameter. This measure is the same as $P@K$ if there are at least K gold labels, otherwise K is reduced to the number of gold labels.

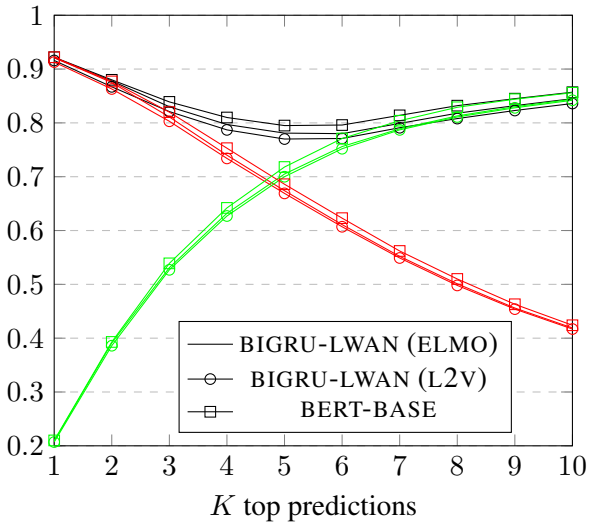

Figure 2: $R@K$ (green lines), $P@K$ (red), $RP@K$ (black) of the best methods (BIGRU-LWAN (L2V), BIGRU-LWAN (ELMO), BERT-BASE), for $K = 1$ to 10.

Figure 2 shows $RP@K$ for the three best systems, macro-averaged over test documents. Unlike $P@K$, $RP@K$ does not decline sharply as K increases, because it replaces K by the number of gold labels, when the latter is lower than K. For $K = 1$, $RP@K$ is equivalent to $P@K$, as confirmed by Fig. 2. For large values of K that almost always exceed the number of gold labels, $RP@K$ asymptotically approaches $R@K$, as also confirmed by Fig. 2.[5] In our dataset, there are 5.07 labels per document, hence $K = 5$ is reasonable.[6]

[5]See Appendix C for a more detailed discussion on the evaluation measures.

[6]Evaluating at other values of K lead to similar conclusions (see Fig. 2 and Appendix D).

Setup: Hyper-parameters are tuned using the HYPEROPT library selecting the values with the best loss on development data.[7] For the best hyper-parameter values, we perform five runs and report mean scores on test data. For statistical significance tests, we take the run of each method with the best performance on development data, and perform two-tailed approximate randomization tests (Dror et al., 2018) on test data.[8] Unless otherwise stated, we used 200-D pretrained GLOVE embeddings (Pennington et al., 2014).

Full documents: The first five horizontal zones of Table 2 report results for full documents. The naive baselines are weak, as expected. Interestingly, for all, frequent, and even few-shot labels, the generic BIGRU-ATT performs better than CNN-LWAN, which was designed for LMTC. HAN also performs better than CNN-LWAN for all and frequent labels. However, replacing the CNN encoder of CNN-LWAN with a BIGRU (BIGRU-LWAN) leads to the best results, indicating that the main weakness of CNN-LWAN is its vanilla CNN encoder.

The zero-shot versions of CNN-LWAN and BIGRU-LWAN outperform all other methods on zero-shot labels (Table 2), in line with the findings of Rios and Kavuluru (2018), because they exploit label descriptors, but more importantly because they have a component that uses prior knowledge as is (i.e., label embeddings are frozen). Exact Match also performs better on zero-shot labels, for the same reason (i.e., the prior knowledge is

[7]We implemented all neural methods in KERAS (https://keras.io/). Code available at https://github.com/iliaschalkidis/lmtc-eurlex57k.git. See Appendix B for details on hyper-parameter tuning.

[8]We perform 10k iterations, randomly swapping in each iteration the responses (sets of returned labels) of the two compared systems for 50% of the test documents.

intact). BIGRU-LWAN, however, is still the best method in few-shot learning. All the differences between the best (bold) and other methods in Table 2 are statistically significant ($p < 0.01$).

Table 3 shows that using WORD2VEC embeddings trained on legal texts (L2V) (Chalkidis and Kampas, 2018) or ELMO embeddings (Peters et al., 2018) trained on generic texts further improve the performance of BIGRU-LWAN.

Document zones: Table 4 compares the performance of BIGRU-LWAN on the development set for different combinations of document zones (Section 3): *header* (*H*), *recitals* (*R*), *main body* (*MB*), full text. Surprisingly *H+R* leads to almost the same results as full documents,[9] indicating that *H+R* provides most of the information needed to assign EUROVOC labels.

	RP@5	$nDCG$@5	Micro-$F1$
GLOVE	0.766	0.796	0.698
L2V	0.775	0.804	0.711
GLOVE + ELMO	0.777	0.808	0.714
L2V + ELMO	**0.781**	**0.811**	**0.719**

Table 3: BIGRU-LWAN with GLOVE, L2V, ELMO.

	μ_{words}	RP@5	$nDCG$@5	Micro-$F1$
H	43	0.747	0.782	0.688
R	317	0.734	0.765	0.669
H+R	360	0.765	0.796	0.701
MB	187	0.643	0.674	0.590
Full	727	**0.766**	**0.797**	**0.702**

Table 4: BIGRU-LWAN with different document zones.

First 512 tokens: Given that *H+R* contains enough information and is shorter than 500 tokens in 83% of our dataset's documents, we also apply BERT to the first 512 tokens of each document (truncated to BERT's max. length), comparing to BIGRU-LWAN also operating on the first 512 tokens. Table 2 (bottom zone) shows that BERT outperforms all other methods, even though it considers only the first 512 tokens. It fails, however, in zero-shot learning, since it does not have a component that exploits prior knowledge as is (i.e., all the components are fine-tuned on training data).

6 Limitations and Future Work

One major limitation of the investigated methods is that they are unsuitable for *Extreme* Multi-Label Text Classification where there are hundreds of thousands of labels (Liu et al., 2017; Zhang et al., 2018; Wydmuch et al., 2018), as opposed to the LMTC setting of our work where the labels are in the order of thousands. We leave the investigation of methods for extremely large label sets for future work. Moreover, RNN (and GRU) based methods have high computational cost, especially for long documents. We plan to investigate more computationally efficient methods, e.g., dilated CNNs (Kalchbrenner et al., 2017) and Transformers (Vaswani et al., 2017; Dai et al., 2019). We also plan to experiment with hierarchical flavors of BERT to surpass its length limitations. Furthermore, experimenting with more datasets e.g., RCV1, Amazon-13K, Wiki-30K, MIMIC-III will allow us to confirm our conclusions in different domains. Finally, we plan to investigate Generalized Zero-Shot Learning (Liu et al., 2018).

[9] The approximate randomization tests detected no statistically significant difference in this case ($p = 0.20$).

Acknowledgements

This work was partly supported by the Research Center of the Athens University of Economics and Business.

References

Nikolaos Aletras et al. 2016. Predicting judicial decisions of the European Court of Human Rights: a Natural Language Processing perspective. *PeerJ Computer Science*, 2:e93.

Ilias Chalkidis, Ion Androutsopoulos, and Achilleas Michos. 2017. Extracting Contract Elements. In *Proceedings of the 16th Edition of the International Conference on Articial Intelligence and Law*, pages 19–28, London, UK.

Ilias Chalkidis, Ion Androutsopoulos, and Achilleas Michos. 2018. Obligation and Prohibition Extraction Using Hierarchical RNNs. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 254–259, Melbourne, Australia.

Ilias Chalkidis and Dimitrios Kampas. 2018. Deep learning in law: early adaptation and legal word embeddings trained on large corpora. *Artificial Intelligence and Law*.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime G. Carbonell, Quoc V. Le, and Ruslan Salakhutdinov. 2019. Transformer-XL: Attentive Language Models Beyond a Fixed-Length Context. *CoRR*, abs/1901.02860.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the Conference of the North American Chapter of the Association for*

Computational Linguistics: Human Language Technologies, Minneapolis, MN, USA.

Rotem Dror, Gili Baumer, Segev Shlomov, and Roi Reichart. 2018. The Hitchhiker's Guide to Testing Statistical Significance in Natural Language Processing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1383–1392, Melbourne, Australia.

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

Nal Kalchbrenner, Lasse Espeholt, Karen Simonyan, Aäron van den Oord, Alex Graves, and Koray Kavukcuoglu. 2017. Neural Machine Translation in Linear Time. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Copenhagen, Denmark.

David D. Lewis, Yiming Yang, Tony G. Rose, and Fan Li. 2004. RCV1: A New Benchmark Collection for Text Categorization Research. *J. Mach. Learn. Res.*, 5:361–397.

Jingzhou Liu, Wei-Cheng Chang, Yuexin Wu, and Yiming Yang. 2017. Deep Learning for Extreme Multi-label Text Classification. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '17, pages 115–124, New York, NY, USA.

Shichen Liu, Mingsheng Long, Jianmin Wang, and Michael I Jordan. 2018. Generalized Zero-Shot Learning with Deep Calibration Network. In *Advances in Neural Information Processing Systems 31*, pages 2005–2015. Curran Associates, Inc.

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

Julian McAuley and Jure Leskovec. 2013. Hidden Factors and Hidden Topics: Understanding Rating Dimensions with Review Text. In *Proceedings of the 7th ACM Conference on Recommender Systems*, RecSys '13, pages 165–172, New York, NY, USA.

Eneldo Loza Mencia and Johannes Fürnkranzand. 2007. An Evaluation of Efficient Multilabel Classification Algorithms for Large-Scale Problems in the Legal Domain. In *Proceedings of the LWA 2007*, pages 126–132, Halle, Germany.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. In *Proceedings of the International Conference on Learning Representations (ICLR)*, Scottsdale, AZ.

James Mullenbach, Sarah Wiegreffe, Jon Duke, Jimeng Sun, and Jacob Eisenstein. 2018. Explainable Prediction of Medical Codes from Clinical Text. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1101–1111, New Orleans, Louisiana. Association for Computational Linguistics.

Ramesh Nallapati and Christopher D. Manning. 2008. Legal Docket Classification: Where Machine Learning Stumbles. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 438–446, Honolulu, Hawaii. Association for Computational Linguistics.

Ioannis Partalas, Aris Kosmopoulos, Nicolas Baskiotis, Thierry Artières, Georgios Paliouras, Éric Gaussier, Ion Androutsopoulos, Massih-Reza Amini, and Patrick Gallinari. 2015. LSHTC: A Benchmark for Large-Scale Text Classification. *CoRR*, abs/1503.08581.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, New Orleans, Louisiana, USA. Association for Computational Linguistics.

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

George Tsatsaronis, Georgios Balikas, Prodromos Malakasiotis, Ioannis Partalas, Matthias Zschunke, Michael R. Alvers, Dirk Weissenborn, Anastasia Krithara, Sergios Petridis, Dimitris Polychronopoulos, Yannis Almirantis, John Pavlopoulos, Nicolas Baskiotis, Patrick Gallinari, Thierry Artières, Axel-Cyrille Ngonga Ngomo, Norman Heino, Éric Gaussier, Liliana Barrio-Alvers, Michael Schroeder, Ion Androutsopoulos, and Georgios Paliouras. 2015. An overview of the BIOASQ large-scale biomedical semantic indexing and question answering competition. *BMC Bioinformatics*, 16(138).

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 *Proceedings of the 31th Annual Conference on Neural Information Processing Systems*, Long Beach, CA, USA.

Marek Wydmuch, Kalina Jasinska, Mikhail Kuznetsov, Róbert Busa-Fekete, and Krzysztof Dembczynski. 2018. A no-regret generalization of hierarchical softmax to extreme multi-label classification. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 6355–6366. Curran Associates, Inc.

Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. 2015. Show, Attend and Tell: Neural Image Caption Generation with Visual Attention. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 2048–2057, Lille, France. PMLR.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical Attention Networks for Document Classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1480–1489, San Diego, California. Association for Computational Linguistics.

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. *CoRR*, abs/1811.01727.

Wenjie Zhang, Junchi Yan, Xiangfeng Wang, and Hongyuan Zha. 2018. Deep Extreme Multi-label Learning. In *Proceedings of the 2018 ACM on International Conference on Multimedia Retrieval*, ICMR '18, pages 100–107, New York, NY, USA.

Arkaitz Zubiaga. 2012. Enhancing Navigation on Wikipedia with Social Tags. *CoRR*, abs/1202.5469.

Appendix

A EURLEX57K statistics

Figure 3 shows the distribution of labels across EURLEX57K documents. From the 7k labels fewer than 50% appear in more than 10 documents. Such an aggressive Zipfian distribution has also been noted in medical code predictions (Rios and Kavuluru, 2018), where such thesauri are used to classify documents, demonstrating the practical importance of few-shot and zero-shot learning.

B Hyper-paramater tuning

Table 5 shows the best hyper-parameters returned by HYPEROPT. Concerning BERT, we set the dropout rate and learning rate to 0.1 and 5e-5, respectively, as suggested by Devlin et al. (2018), while batch size was set to 8 due to GPU memory limitations. Finally, we noticed that the model did not converge in the fourth epoch, as suggested by Devlin et al. (2018). Thus we used early-stopping with no patience and trained the model for eight to nine epochs on average among the five runs.

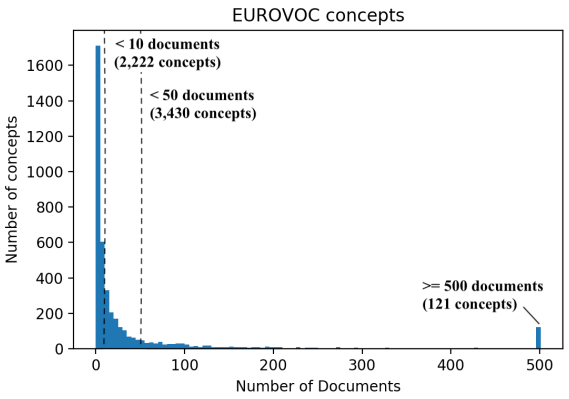

Figure 3: Distribution of EUROVOC concepts across EURLEX57K documents

C Evaluation Measures

The macro-averaged versions of $R@K$ and $P@K$ are defined as follows:

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

$$P@K = \frac{1}{T}\sum_{t=1}^{T}\frac{S_t(K)}{K} \quad (2)$$

where T is the total number of test documents, K is the number of labels to be selected per document, $S_t(K)$ is the number of correct labels among those ranked as top K for the t-th document, and R_t is the number of gold labels for each document. Although these measures are widely used in LMTC, we question their appropriateness for the following reasons:

1. $R@K$ leads to excessive penalization when documents have more than K gold labels. For example, evaluating at $K = 1$ for a single document with 5 gold labels returns $R@1 = 0.20$, if the system managed to return a correct label. The system is penalized, even though it was not allowed to return more than one label.

2. $P@K$ does the same for documents with fewer than K gold labels. For example, evaluating at $K = 5$ for a single document with a single gold label returns $P@1 = 0.20$.

3. Both measures over- or under-estimate performance on documents whose number of gold la-

Hyper parameters	BIGRU-ATT	HAN	CNN-LWAN	BIGRU-LWAN	ZACNN *	ZAGRU *	BERT-BASE +
$N_l \in [1, 2]$	1	(1,1)	1	1	1	1	12
$HU \in [200, 300, 400]$	300	(300,300)	200	300	200	100	768
$D_d \in [0.1, 0.2, \ldots, 0.5]$	0.2	0.3	0.1	0.4	0.1	0.1	0.1
$D_{we} \in [0.00, 0.01, 0.02]$	0.02	0.02	0.01	0.00	0.00	0.00	0.00
$BS \in [8, 12, 16]$	12	16	12	16	16	16	8

Table 5: Best hyper parameters for neural methods. N_l: number of layers, HU: hidden units size, D_d: dropout rate across dimensions, D_{we}: dropout rate of word embeddings, BS: batch size. * Hidden units size is fixed to word embedding dimensionality, + N_l, HU are fixed from the pre-trained model. Dropout rate fixed as suggested by Devlin et al. (2018).

	OVERALL			FREQUENT			FEW			ZERO		
	@1	@5	@10	@1	@5	@10	@1	@5	@10	@1	@5	@10
Exact Match	0.131	0.084	0.080	0.194	0.166	0.141	0.037	0.037	0.036	0.178	0.042	0.022
Logistic Regression	0.861	0.613	0.378	0.864	0.604	0.368	0.458	0.169	0.094	0.011	0.002	0.002
BIGRU-ATT	0.899	0.654	0.407	0.893	0.627	0.382	0.551	0.212	0.121	0.015	0.008	0.007
HAN	0.894	0.643	0.401	0.889	0.620	0.378	0.510	0.199	0.114	0.020	0.011	0.008
CNN-LWAN	0.853	0.617	0.395	0.849	0.596	0.374	0.521	0.204	0.117	0.011	0.007	0.007
BIGRU-LWAN	0.907	0.661	0.414	0.900	0.631	0.387	0.599	0.222	0.124	0.011	0.006	0.006
ZERO-CNN-LWAN	0.842	0.589	0.371	0.837	0.572	0.355	0.447	0.164	0.094	0.202	0.069	0.040
ZERO-BIGRU-LWAN	0.874	0.619	0.386	0.867	0.599	0.367	0.488	0.184	0.107	**0.247**	**0.093**	**0.057**
BIGRU-LWAN (L2V)	0.913	0.669	0.417	0.905	0.639	0.390	0.593	0.219	0.122	0.013	0.007	0.008
BIGRU-LWAN (L2V) *	0.915	0.664	0.413	0.905	0.637	0.387	0.586	0.214	0.120	0.013	0.010	0.010
BIGRU-LWAN (ELMO) *	0.921	0.674	0.419	0.912	0.644	0.391	0.595	0.226	0.127	0.011	0.009	0.007
BERT-BASE *	**0.922**	**0.687**	**0.424**	**0.914**	**0.656**	**0.394**	**0.611**	**0.229**	**0.129**	0.019	0.006	0.007

Table 6: $P@1$, $P@5$ and $P@10$ results on EURLEX57K for all, frequent, few-shot, zero-shot labels. Starred methods use the first 512 document tokens; all other methods use full documents. Unless otherwise stated, GLOVE embeddings are used.

bels largely diverges from K. This is clearly illustrated in Figure 2 of the main article.

4. Because of these drawbacks, both measures do not correctly single out the best methods.

Based on the above arguments, we believe that R-Precision@K ($RP@K$) and $nDCG@K$ lead to a more informative and fair evaluation. Both measures adjust to the number of gold labels per document, without over- or under-estimating performance when documents have few or many gold labels. The macro-averaged versions of the two measures are defined as follows:

$$RP@K = \frac{1}{T}\sum_{t=1}^{T}\frac{S_t(K)}{\min(K, R_t)} \quad (3)$$

$$nDCG@K = \frac{1}{T}\sum_{t=1}^{T}\sum_{k=1}^{K}\frac{2^{S_t(k)}-1}{\log(1+k)} \quad (4)$$

Again, T is the total number of test documents, K is the number of labels to be selected, $S_t(K)$ is the number of correct labels among those ranked as top K for the t-th document, and R_t is the number of gold labels for each document. In the main article we report results for $K = 5$. The reason is that the majority of the documents of EURLEX57K (57.7%) have at most 5 labels. The detailed distributions can be seen in Figure 4.

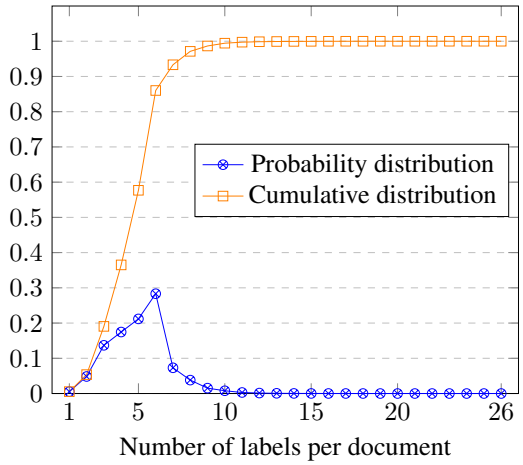

Figure 4: Distribution of number of labels per document in EURLEX57K.

D Experimental Results

In Tables 6–9, we present additional results for the main measures used across the LMTC literature ($P@K$, $R@K$, $RP@K$, $nDGC@K$).

	Overall			Frequent			Few			Zero		
	@1	@5	@10	@1	@5	@10	@1	@5	@10	@1	@5	@10
Exact Match	0.026	0.087	0.168	0.045	0.207	0.344	0.022	0.111	0.214	0.161	0.194	0.206
Logistic Regression	0.195	0.641	0.764	0.234	0.719	0.845	0.313	0.507	0.560	0.011	0.011	0.022
BIGRU-ATT	0.204	0.685	0.824	0.242	0.749	0.880	0.382	0.629	0.703	0.015	0.040	0.062
HAN	0.203	0.675	0.811	0.241	0.740	0.871	0.355	0.596	0.673	0.018	0.051	0.079
CNN-LWAN	0.193	0.647	0.800	0.229	0.713	0.862	0.360	0.612	0.681	0.011	0.036	0.061
BIGRU-LWAN	0.205	0.692	0.836	0.243	0.755	0.891	0.420	0.661	0.725	0.011	0.029	0.060
ZERO-CNN-LWAN	0.189	0.617	0.752	0.223	0.683	0.820	0.300	0.494	0.556	0.189	0.321	0.376
ZERO-BIGRU-LWAN	0.197	0.648	0.782	0.232	0.716	0.847	0.335	0.560	0.635	**0.231**	**0.438**	**0.531**
BIGRU-LWAN (L2V)	0.207	0.700	0.842	0.246	0.764	0.898	0.414	0.655	0.716	0.012	0.034	0.066
BIGRU-LWAN (L2V) *	0.207	0.696	0.835	0.245	0.760	0.891	0.409	0.640	0.707	0.013	0.047	0.084
BIGRU-LWAN (ELMO) *	0.208	0.705	0.844	0.249	0.770	0.900	0.410	0.667	0.732	0.011	0.044	0.061
BERT-BASE *	**0.209**	**0.719**	**0.855**	**0.250**	**0.784**	**0.908**	**0.428**	**0.684**	**0.752**	0.018	0.028	0.068

Table 7: $R@1$, $R@5$ and $R@10$ results on EURLEX57K for all, frequent, few-shot, zero-shot labels. Starred methods use the first 512 document tokens; all other methods use full documents. Unless otherwise stated, GLOVE embeddings are used.

	Overall			Frequent			Few			Zero		
	@1	@5	@10	@1	@5	@10	@1	@5	@10	@1	@5	@10
Exact Match	0.131	0.097	0.168	0.194	0.219	0.344	0.037	0.111	0.214	0.178	0.194	0.206
Logistic Regression	0.861	0.710	0.765	0.864	0.767	0.846	0.458	0.508	0.560	0.011	0.011	0.022
BIGRU-ATT	0.899	0.758	0.824	0.893	0.799	0.880	0.551	0.631	0.703	0.015	0.040	0.062
HAN	0.894	0.746	0.811	0.889	0.789	0.872	0.510	0.597	0.673	0.020	0.051	0.079
CNN-LWAN	0.853	0.716	0.801	0.849	0.761	0.862	0.521	0.613	0.681	0.011	0.036	0.061
BIGRU-LWAN	0.907	0.766	0.836	0.900	0.805	0.891	0.599	0.662	0.725	0.011	0.029	0.060
ZERO-CNN-LWAN	0.842	0.684	0.753	0.837	0.730	0.820	0.447	0.495	0.556	0.202	0.321	0.376
ZERO-BIGRU-LWAN	0.874	0.718	0.782	0.867	0.764	0.847	0.488	0.561	0.635	**0.247**	**0.438**	**0.531**
BIGRU-LWAN (L2V)	0.913	0.775	0.842	0.905	0.815	0.898	0.593	0.657	0.716	0.013	0.034	0.066
BIGRU-LWAN (L2V) *	0.915	0.770	0.836	0.905	0.811	0.891	0.586	0.641	0.707	0.013	0.047	0.084
BIGRU-LWAN (ELMO) *	0.921	0.781	0.845	0.912	0.821	0.901	0.595	0.668	0.732	0.011	0.044	0.061
BERT-BASE *	**0.922**	**0.796**	**0.856**	**0.914**	**0.835**	**0.908**	**0.611**	**0.686**	**0.752**	0.019	0.028	0.068

Table 8: $RP@1$, $RP@5$ and $RP@10$ results on EURLEX57K for all, frequent, few-shot, zero-shot labels. Starred methods use the first 512 document tokens; all other methods use full documents. Unless otherwise stated, GLOVE embeddings are used.

	Overall			Frequent			Few			Zero		
	@1	@5	@10	@1	@5	@10	@1	@5	@10	@1	@5	@10
Exact Match	0.131	0.099	0.134	0.194	0.201	0.262	0.037	0.074	0.112	0.178	0.186	0.189
Logistic Regression	0.861	0.741	0.766	0.864	0.781	0.819	0.458	0.470	0.489	0.011	0.011	0.014
BIGRU-ATT	0.899	0.789	0.819	0.893	0.813	0.853	0.551	0.580	0.608	0.015	0.027	0.034
HAN	0.894	0.778	0.808	0.889	0.805	0.845	0.510	0.544	0.573	0.020	0.034	0.043
CNN-LWAN	0.853	0.746	0.786	0.849	0.772	0.822	0.521	0.557	0.583	0.011	0.023	0.032
BIGRU-LWAN	0.907	0.796	0.829	0.900	0.819	0.861	0.599	0.618	0.643	0.011	0.019	0.029
ZERO-CNN-LWAN	0.842	0.717	0.749	0.837	0.745	0.789	0.447	0.454	0.478	0.202	0.264	0.281
ZERO-BIGRU-LWAN	0.874	0.752	0.781	0.867	0.780	0.819	0.488	0.510	0.539	**0.247**	**0.345**	**0.375**
BIGRU-LWAN (L2V)	0.913	0.804	0.836	0.905	0.828	0.869	0.593	0.612	0.635	0.013	0.024	0.035
BIGRU-LWAN (L2V) *	0.915	0.801	0.832	0.905	0.825	0.864	0.586	0.600	0.625	0.013	0.030	0.042
BIGRU-LWAN (ELMO) *	0.921	0.811	0.841	0.912	0.835	0.874	0.595	0.619	0.643	0.011	0.028	0.034
BERT-BASE *	**0.922**	**0.823**	**0.851**	**0.914**	**0.846**	**0.882**	**0.611**	**0.636**	**0.662**	0.019	0.023	0.036

Table 9: $nDCG@1$, $nDCG@5$ and $nDCG@10$ results on EURLEX57K for all, frequent, few-shot, zero-shot labels. Starred methods use the first 512 document tokens; all other methods use full documents. Unless otherwise stated, GLOVE embeddings are used.