SeCSeq: Semantic Coding for Sequence-to-Sequence based Extreme Multi-label Classification

Wei-Cheng Chang[1]* **Hsiang-Fu Yu**[2] **Inderjit S. Dhillon**[2] **Yiming Yang**[1]
[1] Carnegie Mellon University, [2]Amazon
{wchang2,yiming}@cs.cmu.edu {hsiangfu.isd}@amazon.com

Abstract

Extreme multi-label classification (XMC) aims at assigning to an instance the most relevant subset of labels from a colossal label set. There have been some success in formulating the multi-label problem as sequence-to-sequence (Seq2Seq) learning, where the positive class labels of each input instance are used as the corresponding output sequence. Seq2Seq methods, nonetheless, have not yet been scalable to the XMC setting due to the softmax bottleneck. In this paper, we propose a semantic coding framework, namely SeCSeq, for a Seq2Seq approach to the XMC problem. To circumvent the softmax bottleneck, SeCSeq compresses labels into sequences of semantic-aware compact codes, on which Seq2Seq models are trained. For inference, the generated semantic codes are then decompressed into sequences of positive labels using ensemble techniques. Preliminary experiments on XMC benchmark datasets show that SeCSeq is competitive with the state-of-the-art while requiring significantly fewer model parameters.

1 Introduction

Extreme multi-label classification (XMC) refers to the problem of assigning to an instance the most relevant subset of labels from an enormous label collection, where the number of labels could be in the millions or more. XMC setting is universal in various industrial applications such as Youtube recommendation [2], Amazon query ranking for sponsored product advertising [8], Wikipedia categories tagging in the PASCAL Large-Scale Hierarchical Text Classification (LSHTC) challenge [11], to name just a few. The huge label space raises research challenges such as data sparsity and scalability for the existing multi-label algorithms.

Formally, multi-label classification (MLC) is the task of learning a function f that maps an input $\mathbf{x} \in \mathcal{X}$ to its target $\mathbf{y} \in \mathcal{Y} = \{0,1\}^L$, where L is the number of total unique labels. The objective is to maximize the joint probability of labels conditioned on the input $\mathbf{x}$:

$$\max_f P(y_1, y_2, \ldots, y_L | \mathbf{x}; f) = \prod_{l=1}^{L} P(y_l | \mathbf{x}, y_1, \ldots, y_{l-1}; f). \tag{1}$$

The objective (1) can be interpreted as maximizing the *subset* $0/1$ accuracy, which is the Bayes optimal prediction [5]. We can solve (1) by fitting L conditional independent classifiers as done by the Probabilistic Classifier Chain [5]. While enjoying strong statistical guarantees, PCC is not practical because of the infeasible $O(2^L)$ complexity and the cascading errors during inference. To overcome the above issues, [10] proposes to maximize the joint probability of positive labels:

$$\max_f P(\mathbf{y}_{p_1}, \mathbf{y}_{p_2}, \ldots, \mathbf{y}_{p_T} | \mathbf{x}; f) = \prod_{t=1}^{T} P(\mathbf{y}_{p_t} | \mathbf{x}, \mathbf{y}_1, \ldots, \mathbf{y}_{p_{t-1}}; f), \tag{2}$$

[*] Work performed while at A9.com, an Amazon subsidiary

32nd Conference on Neural Information Processing Systems (NIPS 2018), Montréal, Canada.

where $\mathbf{y}$ can be viewed as a set of 1-of-L vectors $\mathbf{y} = \{\mathbf{y}_{p_t}\}_{t=1}^{T}$, $\mathbf{y}_{p_t} \in \mathbb{R}^L$, and $T \ll L$ is the number of positive labels of the input $\mathbf{x}$. Since instances may have different values of T, [10] considers Seq2Seq models for solving (2). Inference requires computations over L^T search space, which can be approximated by beam search. Although reducing the complexity from $O(2^L)$ to $O(L^T)$ thanks to the sparse label cardinality in XMC, the number of parameters in the softmax layer of Seq2Seq models f grows linearly with L, which hinders the feasibility of Seq2Seq models on industrial XMC applications, where L is the order of millions or more. This stimulates us with the following question: *Can we find a compact representation of labels that has not only a constant sequence length which is smaller than L and but a compact output space independent of L?*

In this paper, we propose SeCSeq, a semantic coding framework for Seq2Seq approaches on the XMC problem. To circumvent the softmax bottleneck, SeCSeq compresses labels into sequences of semantic-aware compact codes, on which Seq2Seq models are trained. For inference, the generated semantic codes are then decompressed to sequences of positive labels using ensemble trees. Preliminary experiments on XMC benchmark datasets not only show SeCSeq has competitive accuracy among the state-of-the-art methods, but also enjoys a compact model size that is constant w.r.t the number of labels.

2 SeCSeq: A Semantic Coded Seq2Seq Framework

In the proposed framework, each label y_l is compressed into a K-way D-dimensional discrete code. We denote the discrete code for the l-th label as $\mathbf{c}_l = \{c_l^1, c_l^2, \dots, c_l^D\} \in \mathcal{B}^D$, where $\mathcal{B} = \{1, 2, \dots, K\}$ is the set of code bits of K cardinality. The goal is to find a compression function $\phi(\cdot) : L \rightarrow \mathcal{B}^D$ and a decompression function $\psi(\cdot) : \mathcal{B}^D \rightarrow L$ with following properties: 1) similar codes should reflect *semantically similar* labels; 2) compression and decompression should be *efficient*; 3) maintain low collision errors while having high compression rate. The design of ϕ and ψ is deferred to the end of this section.

Given an instance $\mathbf{x}$ and its target output $\mathbf{y} = \{\mathbf{y}_{p_t}\}_{t=1}^{T}$, by applying ϕ sequentially on all postive labels $\mathbf{y}_{p_t}$, the compressed sequence is $\{\mathbf{c}_{p_1}, \dots, \mathbf{c}_{p_t}, \dots, \mathbf{c}_{p_T}\}$ where $\mathbf{c}_{p_t} = \{c_{p_t}^1, \dots, c_{p_t}^D\}$ is the K-way D-dimensional code. Thus, the objective becomes

$$\max_f P\big(\{c_{p_1}^1, \dots, c_{p_1}^D\}, \dots, \{c_{p_T}^1, \dots, c_{p_T}^D\} | \mathbf{x}; f\big), \tag{3}$$

which can be solved by Seq2Seq models, as shown in Fig 1. The main intuition is to flexibly control the trade-off between compressed sequence length $T \cdot D$ and the output cardinality K. In the short sequence extreme ($T \cdot D = T, K = L$), we reduce to the mlc2seq objective (2) [10] while for the long sequence extreme ($T \cdot D = L, K = 2$), we reduce to the PCC objective (1) [5]. Another main advantage is that the compact code output space K in the softmax does not explicitly depend on L.

Label Compression Provided the label embedding $\mathbf{v}_l \in \mathbb{R}^P$ of label l, an example realization of $\phi = q \circ r$ is through a random projection r compositing with some quantization functions q [7]

$$\phi_d(\mathbf{v}_l; K) = q\big(r(\mathbf{v}_l)_d; K\big), \ \mathbf{c}_l = \{\phi_1(\mathbf{v}_l; K), \dots, \phi_D(\mathbf{v}_l; K)\}, \ d = 1, \dots, D. \tag{4}$$

Notably, $\mathbf{z} = r(\mathbf{v}_l) = \mathbf{v}_l^T R$ is the random projection mapping where $R \in \mathbb{R}^{P \times D}$, $R_{ij} \overset{iid}{\sim} \mathcal{N}(0, 1)$. Also, q is some quantization function such as uniform quantization or ordinal ranking quantization.

In practice, we sample multiple random projection matrices $R_1, \dots, R_m \in \mathbb{R}^{P \times D}$, and posit the compact code $\mathbf{c}$ as the path traversing the random trees. Figure 2 demonstrates number of trees $m = 2$. Analogous to the random forest ensemble prediction from all trees, SeCSeq embraces similar benefit that m random projection trees encoding are trained by m different Seq2Seq models.

Our framework can readily incorporate the rich label side information, which is abundant in real-world industrial applications. For example, label co-occurrence graph can be transformed into label embedding via various label propagation methods such as DeepWalk [12]. Besides, labels often associated with text description, which can be presented by some low-dimension contextual embedding such as FastText [4] or ELMo [13].

Semantic Code Decompression Suppose we now have a sequence of compressed code $s = \{\{c_{p_1}^1, \dots, c_{p_1}^D\}, \dots, \{c_{p'_T}^1, \dots, c_{p'_T}^D, \} \dots\}$ generated from the Seq2Seq model via Monte Carlo sampling or beam search. To decompress the discrete code to labels, we consider the following greedy scheme: 1) Segment the sequence s into T' chunks, where $T' = \lfloor |s|/D \rfloor$, ignoring the remaining codes at the end; 2) For each chunk of length D subsequences $\{s_{(t-1)D+1}, \dots, s_{tD}\}$, traverse this subsequence along the random projection tree to a leaf node; 3) At the leaf node is a label count histogram, which could be empty; 4) Accumulate the label count histogram for every chunk, and output final label count histogram.

To ensemble prediction from M different random projection trees, we simply aggregate the label histogram count from each tree, and output the top k label with highest counts.

3 Experiment

We compare the proposed SeCSeq to state-of-the-art XMC methods including tree-based FastXML [14], OVA-based PD-Sparse [16], and Seq2Seq-based mlc2seq [10] on public available benchmark multi-label datasets [15]. We follow [10] to obtain tokenized text representation for deep learning methods (mlc2seq and SeCSeq) and use TF-IDF unigram features for feature-based methods (PD-Sparse and FastXML). The data statistics are summarized in Table 3. Details are available in Appendix A. We evaluate all methods in terms of example-based ranking measures (Precision@k, for $k = 1, 3, 5$).

Experiment results of Wiki10-28K and AmazonCat-13K are presented in Table 1 and Table 2, respectively. When the number of random projection tree $M = 1$, the proposed framework SeCSeq can achieve comparable ranking performance as FastXML and PD-Sparse with a more compact model size. For AmazonCat-13K, in particular, the model size of SeCSeq is 0.1 and 0.003 the sizes of the PD-Sparse and FastXML models, respectively. Moreover, we find significant gain as the number of random projection trees from $M = 1$ to $M = 4$, reaching the best on Wiki10-28K and the best P@1,3 on AmazonCat-13K. This initial result encourages us to further explore random projection using different feature embedding such that a more diverse semantic code could potentially improve our framework.

Method	Model Size	Prec@1(%)	Prec@3(%)	Prec@5(%)
FastXML [14]	73M	83.22	68.79	58.48
PD-Sparse [16]	82M	82.36	68.98	57.77
mlc2seq [10]	239M	81.75	65.45	54.18
SeCSeq ($M = 1$)	**46M**	81.61	67.32	56.36
SeCSeq ($M = 4$)	184M	**83.54**	**70.06**	**59.40**

Table 1: results on Wiki10-28K. M is the number of random projection trees.

Method	Model Size	Prec@1(%)	Prec@3(%)	Prec@5(%)
FastXML [14]	15G	92.68	77.21	**62.07**
PD-Sparse [16]	463M	89.18	69.95	55.48
mlc2seq [10]	180M	91.52	74.77	59.13
SeCSeq ($M = 1$)	**46M**	92.10	74.78	59.05
SeCSeq ($M = 4$)	184M	**93.19**	**77.34**	61.74

Table 2: results on AmazonCat-13K. M is the number of random projection trees.

4 Conclusions

In this paper, we propose SeCSeq, a semantic coding framework for Seq2Seq approaches to the XMC problem. To circumvent the softmax bottleneck, SeCSeq compresses labels into sequences of semantic-aware yet compact codes, on which Seq2Seq models are trained. For inference, the generated semantic codes are then decompressed to sequences of positive labels using ensemble trees. Preliminary experiments on XMC benchmark datasets show that SeCSeq is competitive with the state-of-the-art while requiring significantly fewer model parameters. Future work will focus on the study of different quantization techniques and incorporation of heterogeneous label embeddings (label graph embedding, ELMO, etc).

References

[1] Rohit Babbar and Bernhard Schölkopf. Dismec: distributed sparse machines for extreme multi-label classification. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, pages 721–729. ACM, 2017.

[2] Alex Beutel, Paul Covington, Sagar Jain, Can Xu, Jia Li, Vince Gatto, and Ed H Chi. Latent cross: Making use of context in recurrent recommender systems. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, pages 46–54. ACM, 2018.

[3] Kush Bhatia, Himanshu Jain, Purushottam Kar, Manik Varma, and Prateek Jain. Sparse local embeddings for extreme multi-label classification. In *Advances in Neural Information Processing Systems*, pages 730–738, 2015.

[4] 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. ISSN 2307-387X.

[5] Krzysztof Dembczynski, Weiwei Cheng, and Eyke Hüllermeier. Bayes optimal multilabel classification via probabilistic classifier chains. In *ICML*, volume 10, pages 279–286, 2010.

[6] Himanshu Jain, Yashoteja Prabhu, and Manik Varma. Extreme multi-label loss functions for recommendation, tagging, ranking & other missing label applications. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 935–944. ACM, 2016.

[7] Ping Li, Michael Mitzenmacher, and Anshumali Shrivastava. Coding for random projections. In *International Conference on Machine Learning*, pages 676–684, 2014.

[8] Greg Linden, Brent Smith, and Jeremy York. Amazon. com recommendations: Item-to-item collaborative filtering. *IEEE Internet computing*, (1):76–80, 2003.

[9] Julian McAuley and Jure Leskovec. Hidden factors and hidden topics: understanding rating dimensions with review text. In *Proceedings of the 7th ACM conference on Recommender systems*, pages 165–172. ACM, 2013.

[10] Jinseok Nam, Eneldo Loza Mencía, Hyunwoo J Kim, and Johannes Fürnkranz. Maximizing subset accuracy with recurrent neural networks in multi-label classification. In *Advances in Neural Information Processing Systems*, pages 5413–5423, 2017.

[11] Ioannis Partalas, Aris Kosmopoulos, Nicolas Baskiotis, Thierry Artieres, George Paliouras, Eric Gaussier, Ion Androutsopoulos, Massih-Reza Amini, and Patrick Galinari. Lshtc: A benchmark for large-scale text classification. *arXiv preprint arXiv:1503.08581*, 2015.

[12] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 701–710. ACM, 2014.

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

[14] Yashoteja Prabhu and Manik Varma. Fastxml: A fast, accurate and stable tree-classifier for extreme multi-label learning. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 263–272. ACM, 2014.

[15] Manik Varma. The extreme classification repository: Multi-label datasets & code. http://manikvarma.org/downloads/XC/XMLRepository.html, 2018. Accessed: 2018-10-5.

[16] Ian EH Yen, Xiangru Huang, Kai Zhong, Pradeep Ravikumar, and Inderjit S Dhillon. Pd-sparse: A primal and dual sparse approach to extreme multiclass and multilabel classification. In *Proceedings of the 33th International Conference on Machine Learning*, 2016.

[17] Arkaitz Zubiaga. Enhancing navigation on wikipedia with social tags. In *Wikimania 2009*. Wikimedia Foundation, 2009.

A Datasets and Preprocessing

Dataset	N_{trn}	N_{val}	N_{tst}	#features	#labels
Wiki10-28K	11,265	1,251	5,732	99,920	28,139
AmazonCat-13K	1,067,616	118,623	306,782	161,926	13,234

Table 3: Data Statistics. $N_{trn}, N_{val}, N_{tst}$ refer to number of instances in training, validation, and test set, respectively. The dataset is denote as its name, followed by a dash sign, followed by the #labels.

We consider two multi-label text classification datasets downloaded from the publicly available Extreme Classification Repository [15] for which we had access to the full text, namely Wiki10-28K and AmazonCat-13K. Summary statistics of the datasets are given in Table 3. We follow the training and test split of [15] and set aside 10% of the training instances as the validation set for hyperparameter tunning. However, since we adhere the text preprocessing procedure of [10], the data statistics, particularly the #labels, would be slightly different from [15]. Specifically, for raw text input to Seq2Seq-based methods,we replaced numbers with a special token and then build a word vocabulary using the most frequent 50K words. Out-of-vocabulary (OOV) words were also replaced with a special token and we truncated the documents after 300 words.

Wiki10-28K This dataset is originally from [17], made up by $20,764$ unique English Wikipedia articles with at least 10 annotations from the social bookmarking website Delicious. Wiki10-28K is widely used in the XMC literature such as SLEEC [3], PfastreXML [6] and DiSMEC [1].

AmazonCat-13K This dataset is originally from [9], made up by the co-purchase review data, along with the ASIN title, descriptions, and categories. AmazonCat-13K is used in the XMC literature such as PfastreXML [6] and DiSMEC [1].

B Model Architecture

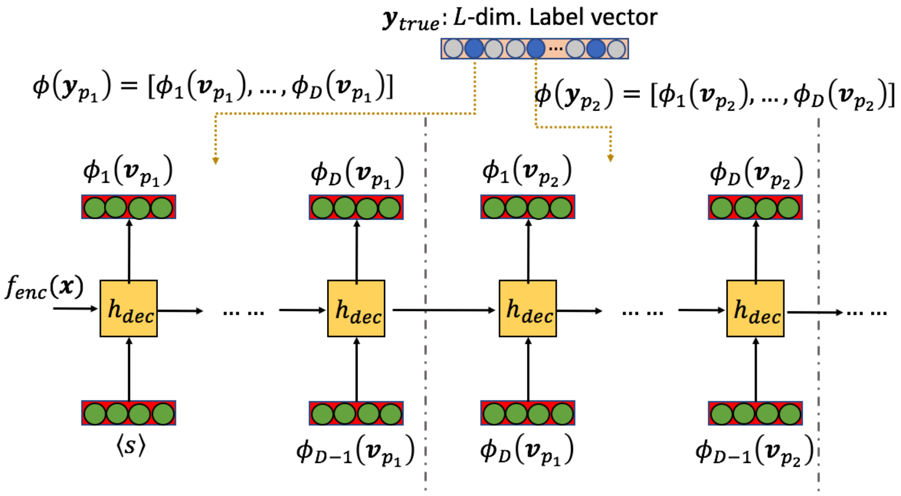

Figure 1: The proposed SeCSeq framework. Each positive label $\mathbf{y}_{p_t}$ is represented as a K-way D-dimensional semantic code $\mathbf{c}_{p_t} = \{c^1_{p_t}, \dots, c^D_{p_t}\}$ via some label compression functions ϕ. We concatenate all positive labels' codes into a longer sequence for training. The softmax layer now becomes $O(K)$, which significantly reduces model size, speedup training and inference time.

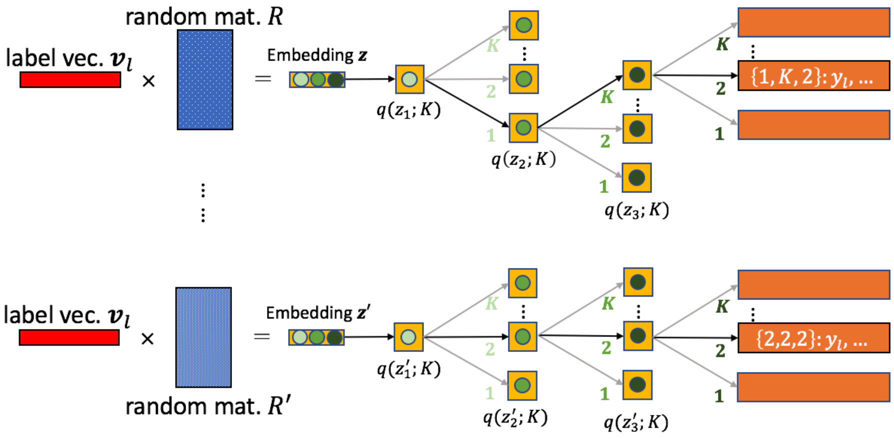

Figure 2: an instance of $\phi(\cdot)$ function using random projection trees. Given a label embedding $\mathbf{v}_l$, several semantic-aware embedding are available such as instance indicator vector, word2vec embedding, Fasttext embedding, and more. Furthermore, we may consider an ensemble of M random projection trees, that leads to a set of M different semantic codes, and train the SeCSeq in parallel without any further cost.