Pretrained Generalized Autoregressive Model with Adaptive Probabilistic Label Clusters for Extreme Multi-label Text Classification

Hui Ye[1], Zhiyu Chen[1], Da-Han Wang[2], Brian D. Davison[1]

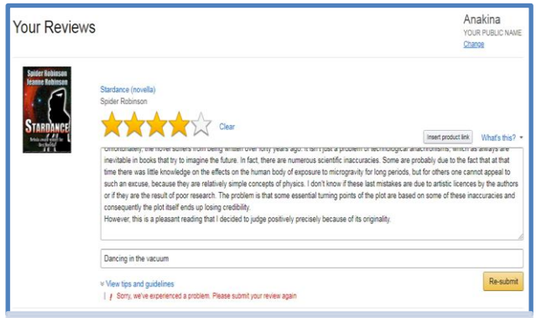
ICML 2020

Extreme Multi-label Text Classification

Binary Classification

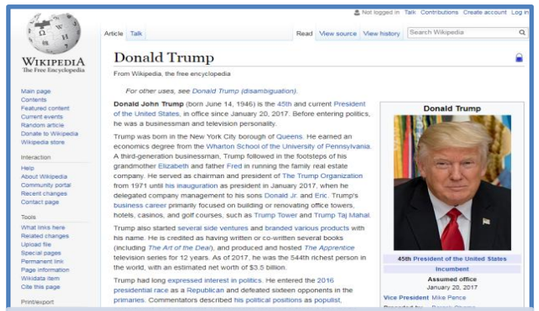
- spam
- not spam

Multi-class Classification



- 1 star
- 2 star
- 3 star
- 4 star
- 5 star

Multi-label Classification



- politician
- businesspeople
- musician
- engineer
- scientist
- …

How to deal with extreme outputs efficiently

- **Tree-based approaches**
  - Train a hierarchical tree structure for fast training and prediction

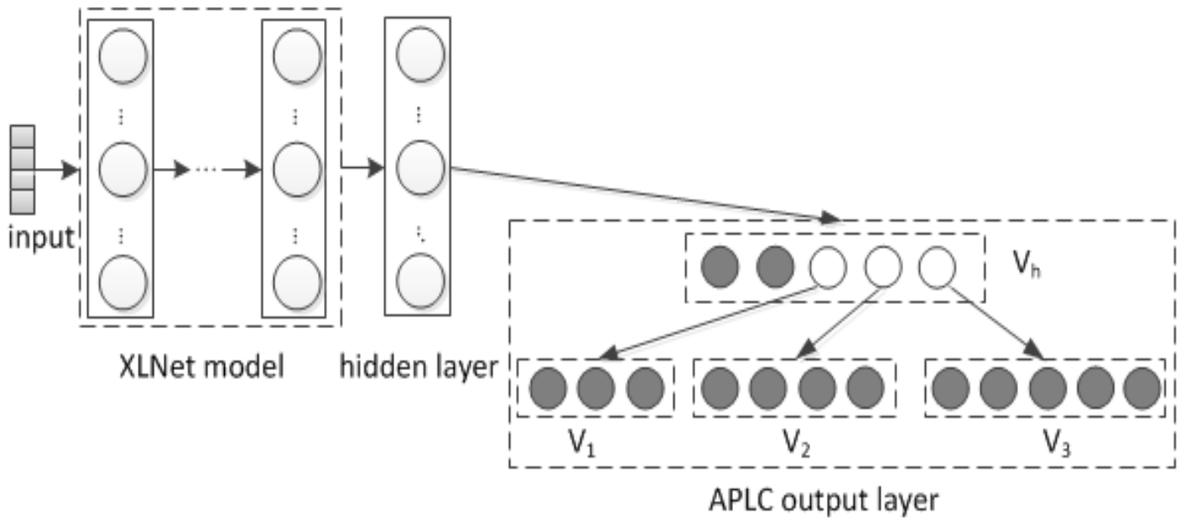
- **Embedding-based approaches**
  - Project high-dimensional label space into low-dimensional one

- **Sampling approaches**
  - Sample a small percentage of negative labels

- **Cluster-based approaches**
  - Partition the label set into several clusters

Model Architecture

- **Backbone model: XLNet (Dai et al., 2019)**
  - Generalized autoregressive pretraining language model
  - Fine-tune the model on downstream tasks

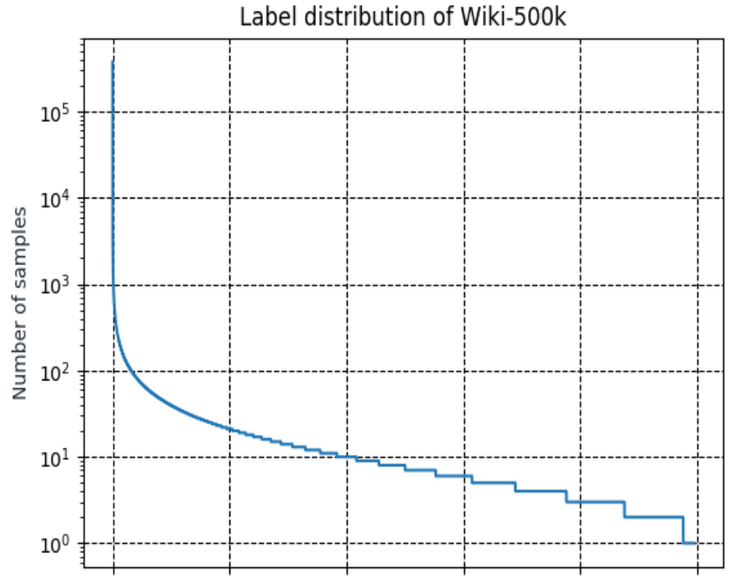


- **Hidden layer**
  - Between XLNet and APLC output layer
  - Bottleneck layer : reduce 768 to 512, 256 et al.

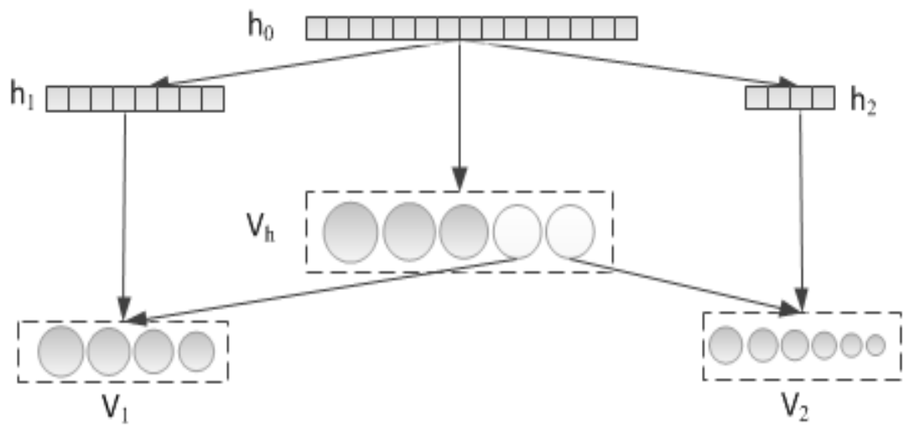
- **Output layer : Adaptive Probabilistic Label Clusters (APLC)**

Adaptive Probabilistic Label Clusters

- **Motivation: Zipf's Law**
  - Most of the probability mass is covered by only a small fraction of the label set
  - i.e. 25% labels cover 75% probability mass



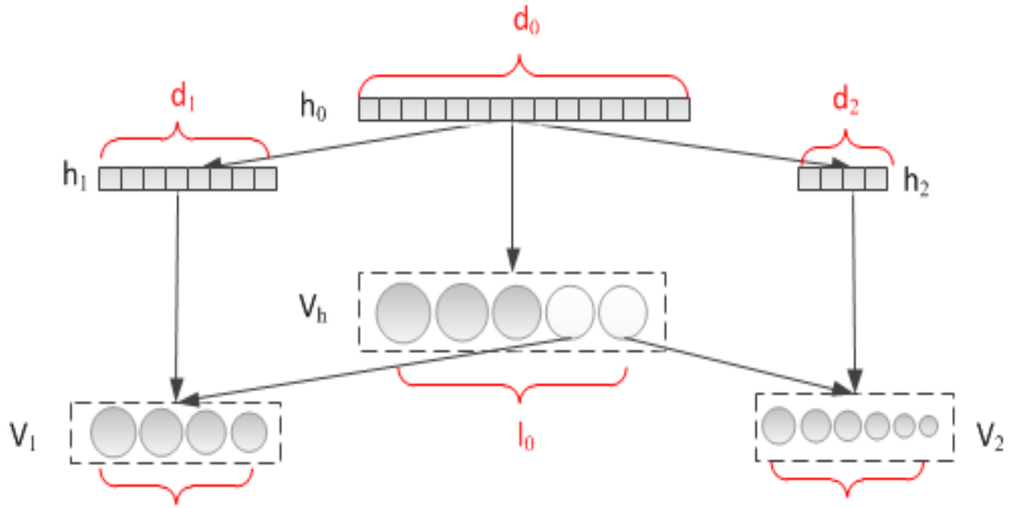
- **Architecture**
  - Most frequent labels in head cluster, infrequent labels in tail clusters
  - Keep the head cluster as a short-list in the root node
  - Decreasing dimension of hidden state: $q = \frac{d_i}{d_{i-1}},\ q \geq 1$



Adaptive Probabilistic Label Clusters

- **Model size**:
  - Partition of the label set : v_h, v_1, v_2
  - Decay factor $q = \frac{d_i}{d_{i-1}},\ q \geq 1$

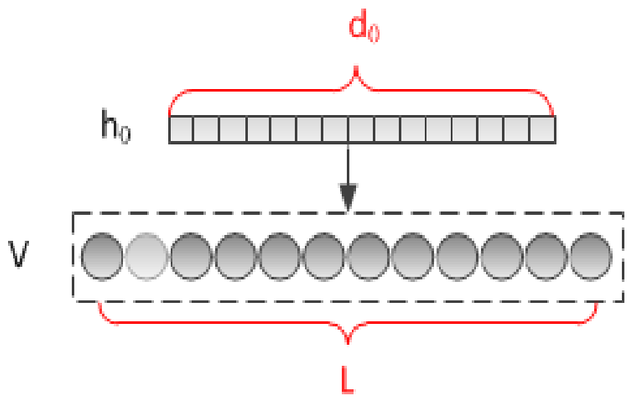
$$N_{par} \approx dl_h + \sum_{i=1}^{K} \frac{d}{q^i} l_i = d \sum_{i=0}^{K} \frac{l_i}{q^i} \qquad (7)$$



APLC

- **How the model size is reduced**
  - i.e. 3 clusters, q = 2,

$$\underbrace{(d*l_0 + \frac{d}{2}*l_1 + \frac{d}{4}*l_2)}_{\text{APLC}} < \underbrace{(d*l_0 + \mathrm{d}*l_1 + \mathrm{d}*l_2)}_{\text{Linear Output}} = \mathrm{d}*\mathrm{L}$$

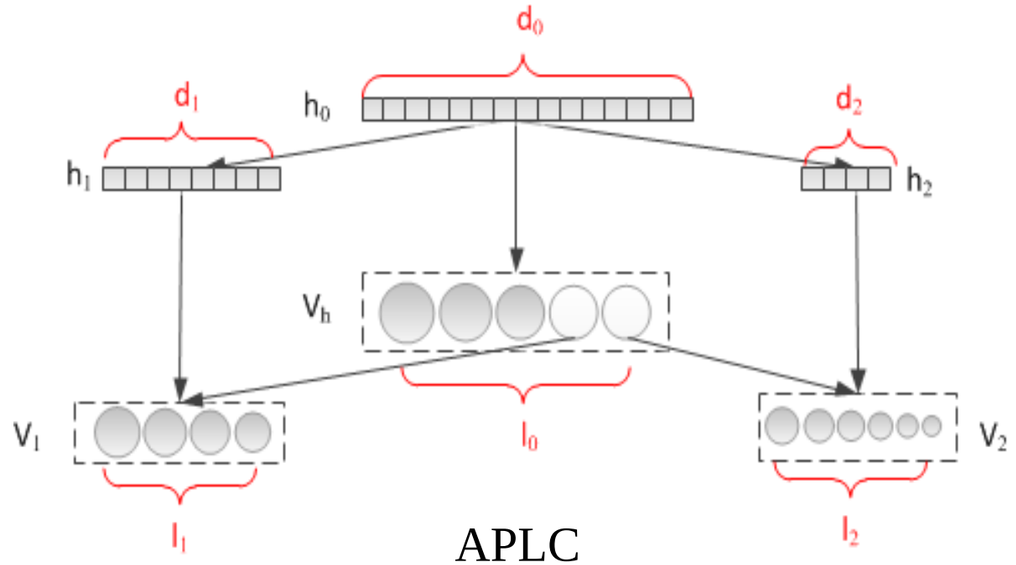


Linear output

Adaptive Probabilistic Label Clusters

- **Computational cost**：
  - Model visits cluster v_i with probability p_i

$$C = O(N_b d(l_h + \sum_{i=1}^{K} p_i \frac{l_i}{q^i})) \qquad (12)$$



APLC

- **How the computational cost is reduced**
  - i.e. 3 clusters , $(x_i, y_i) = (x_i, [y_i^1, y_i^2, y_i^3])$ has 3 positive labels

Table shows the clusters that the model visits

		Linear Structure	**APLC**	
Case 1	$y_i^1, y_i^2, y_i^3 \in V_h$	V_h, V_1, V_2	V_h	best
Case 2	$y_i^1, y_i^2 \in V_h, y_i^3 \in V_1$	V_h, V_1, V_2	V_h , V_1	
Case 3	$y_i^1 \in V_h, y_i^2 \in V_1, y_i^3 \in V_2$	V_h, V_1, V_2	V_h, V_1, V_2	worst
…	…	…	…	

Datasets

Table 1. Statistics of datasets. N_{train} is the number of training samples, N_{test} is the number of test samples, D is the dimension of the feature vector, L is the cardinality of the label set, $\bar{L}$ is the average number of labels per sample, $\hat{L}$ is the average samples per label, $\overline{W}_{train}$ is the average number of words per training sample and $\overline{W}_{test}$ is the average number of words per test sample.

Dataset	N_{train}	N_{test}	D	L	$\bar{L}$	$\hat{L}$	$\overline{W}_{train}$	$\overline{W}_{test}$
EURLex-4k	15,449	3,865	186,104	**3,956**	5.30	20.79	1,248.58	1,230.40
AmazonCat-13k	1,186,239	306,782	203,882	**13,330**	5.04	448.57	246.61	245.98
Wiki10-31k	14,146	6,616	101,938	**30,938**	18.64	8.52	2,484.30	2,425.45
Wiki-500k	1,779,881	769,421	2,381,304	**501,008**	4.75	16.86	808.66	808.56
Amazon-670k	490,449	153,025	135,909	**670,091**	5.45	3.99	247.33	241.22

Implementation details of APLC

Table 2. Implementation details of APLC. d_h is the dimension of the input hidden state, q is the factor by which the dimension of hidden state for the tail cluster decreases, N_{cls} is the number of clusters and P_{num} is the proportion for which the number of labels in each cluster accounts.

Dataset	d_h	q	N_{cls}	P_{num}
EURLex-4k	768	2	2	0.5, 0.5
AmazonCat-13k	768	2	2	0.5, 0.5
Wiki10-31k	768	2	2	0.5, 0.5
Wiki-500k	768	2	3	0.33, 0.33, 0.34
Amazon-670k	512	2	4	0.25, 0.25, 0.25, 0.25

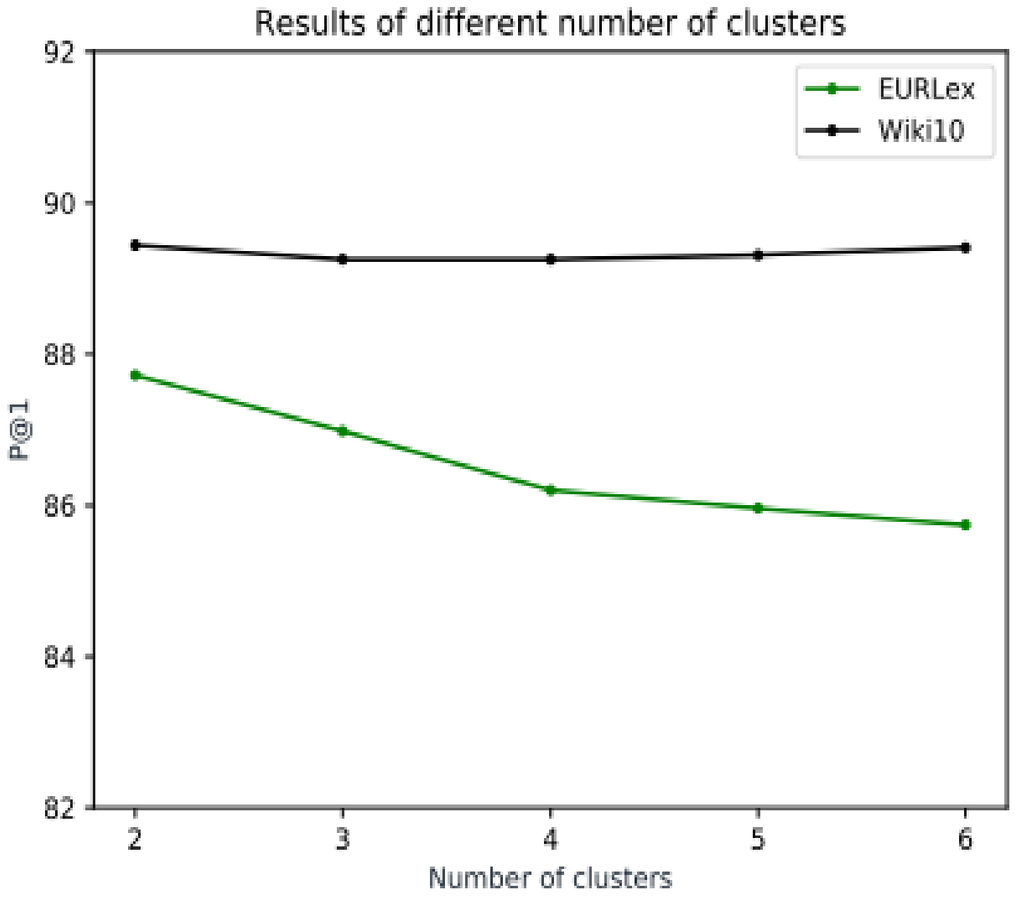
Results

- **Metric** : Precision at 1, 3, 5

Dataset		SLEEC	AnnexML	DisMEC	PfastreXML	Parabel	Bonsai	XML-CNN	AttentionXML	APLC-XLNet
EURLex-4k	P@1	79.26	79.66	82.40	75.45	81.73	83.00	76.38	87.14	**87.72**
	P@3	64.30	64.94	68.50	62.70	68.78	69.70	62.81	**75.18**	74.56
	P@5	52.33	53.52	57.70	52.51	57.44	58.40	51.41	**62.58**	62.28
AmazonCat-13k	P@1	90.53	93.55	93.40	91.75	93.03	92.98	93.26	92.62	**94.56**
	P@3	76.33	78.38	79.10	77.97	79.16	79.13	77.06	77.56	**79.82**
	P@5	61.52	63.32	64.10	63.68	64.52	64.46	61.40	62.74	**64.60**
Wiki10-31k	P@1	85.88	86.50	85.20	83.57	84.31	84.70	84.06	86.04	**89.44**
	P@3	72.98	74.28	74.60	68.61	72.57	73.60	73.96	77.54	**78.93**
	P@5	62.70	64.19	65.90	59.10	63.39	64.70	64.11	68.48	**69.73**
Wiki-500k	P@1	53.60	63.86	70.20	56.25	68.52	69.20	59.85	72.62	**72.83**
	P@3	34.51	40.66	50.60	37.32	49.42	49.80	39.28	**51.02**	50.50
	P@5	25.85	29.79	**39.70**	28.16	38.55	38.80	29.81	39.41	38.55
Amazon-670k	P@1	35.05	42.08	44.70	39.46	44.89	**45.50**	35.39	45.45	43.46
	P@3	31.25	36.65	39.70	35.81	39.80	40.30	31.93	**40.63**	38.82
	P@5	28.56	32.76	36.10	33.05	36.00	36.50	29.32	**36.92**	35.32

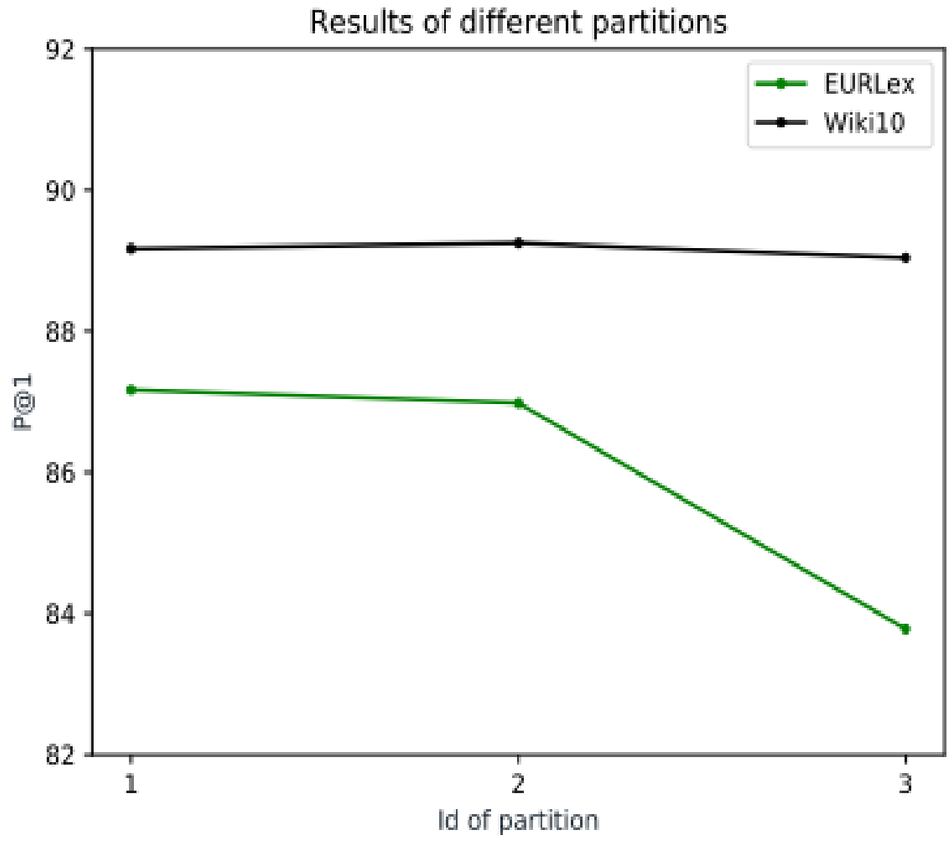
Ablation study

- Impact of the number of clusters



- Impact of the partition of the label set

(0.7,0.2,0.1), (0.33,0.33,0.34) and (0.1,0.2,0.7) for (P_h, P_1, P_2)



Summary

- Proposed a novel deep learning approach for the XMTC problem
- Proposed APLC to deal with extreme labels efficiently
- Carried out theoretical analysis on APLC
  - Model size
  - Computational cost