Classification

CLS 4. Extreme-scale Neural Classifiers

Outline

- Introduction to extreme-scale classification
- Leveraging label dependencies (hierarchical or graphical) in regularization [SIGKDD 2013]
- Neural models with label clustering for extreme-scale multi-label classification [NIPS 2019; SIGKDD 2020]

Extreme-scale Classification Problems

- Wikipedia articles → 1 million curator-generated categories (in a connected graph)
- Amazon products → 2.8 million categories of videos, books, computers, software, clothing, jewelries, …
- Amazon products reviews → 670k social tags (by users)
- Medical journal articles → 20k Medical Subject Headings (hierarchy)

⋮

Wikipedia Page of ChatGPT

ChatGPT, which stands for **Chat Generative Pre-trained Transformer**, is a large language model-based chatbot developed by OpenAI and launched on November 30, 2022, that enables users to refine and steer a conversation towards a desired length, format, style, level of detail, and language. Successive prompts and replies, known as prompt engineering, are considered at each conversation stage as a context.[2]…

Categories:

- Categories: ChatGPT
- OpenAI
- Chatbots
- Large language models
- Generative pre-trained transformers
- Interactive narrative
- Virtual assistants
- Applications of artificial intelligence
- 2022 software

Amazon Product Review

5

Multi-class Benchmark Datasets
(each instance has one and only one label)

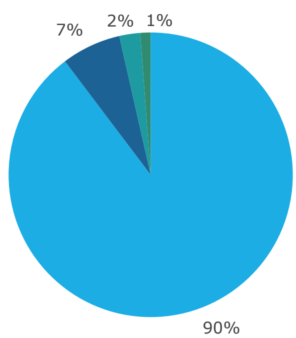
Dataset	Data Type	# of Categories	# of Features (e.g., Words)	# of Training Instances	# of Test Instances
IMBD	Sentiment	2	438,729	25,000	25,000
Yelp	Reviews	5	171,846	650,000	50,000
Cifar-10	Images	10	3,072	5,000	1,000
NEWS20	News stories	20	53,975	11,260	7,505
RCV1	News stories	137	48,734	23,149	784,446
IPC	Patents	552	541,869	46,324	28,926
LSHTC-small	Web pages	1,563	51,033	4,463	1,858
ImageNet	Images	21,841	4,096	12,777,400	1,419,712
LWIKI 2011	Wikipedia	478,020	1,617,899	2,365,436	452,167

6

Multi-label Benchmark Datasets
(each instance can have more than one labels)

Dataset (Type)	Label Type	# of Categories	# of Features	# of Training Instances	# of Test Instances	# of Labels per Instance
EURLex-4K	Legal categories	3,993	186,104	15,539	3,809	5.31
Wiki10-31K	Social tags	30,938	101,938	14,146	6,616	18.64
Amazon-13K (product description)	Product categories	13,330	203,882	1,186,239	306,782	5.04
Wiki-500K	Topics	501,008	2,381,304	1,779,881	769,421	4.75
Amazon-670K (Review)	Product IDs	670,091	135,909	490,449	153,025	5.45
Amazon-3M (product description)	Product categories	2,812,281	337,067	1,717,899	742,507	36.04

Skewed Category Distribution (e.g., Wiki10-31K)

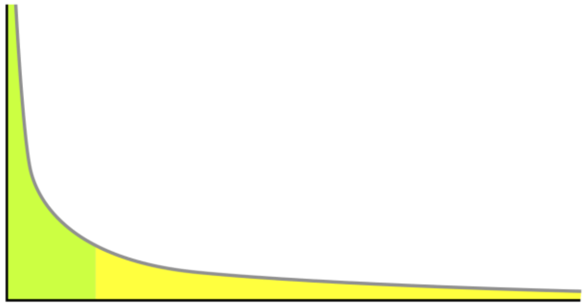


#instances

1-10	90%
11-30	7%
31-90	2%
>=91	1%

- 90% of the categories (blue) has 1-10 instances;
- 1% of the categories (purple) has >90 instances.

The Power Law Phenomena
(https://en.wikipedia.org/wiki/Power_law)



An example power-law graph that demonstrates ranking-vs-frequency property. To the right (yellow) is the long tail, and to the left (green) are the dominating ones (also known as the 80–20 rule).

The Power Law

- The relationship between x and y in an exponential form

$$y = cx^a \qquad (a \text{ and c are some constants})$$

- The relationship between x and y is linear in the log-scale

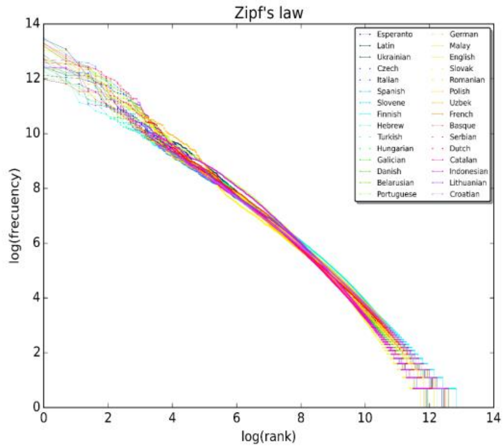
$$\underbrace{y'}_{\log y} = a \underbrace{x'}_{\log x} + \text{constant}$$

- Zipf's law is a special case of the power law ($a = -1$)

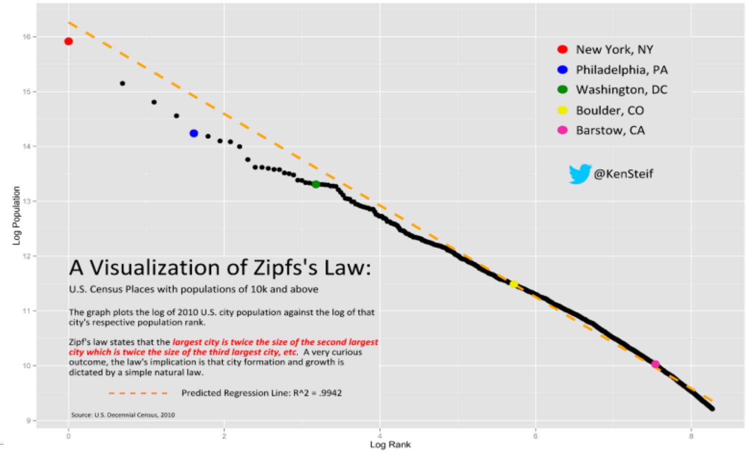
$$y \propto \frac{1}{x} \quad \Rightarrow \quad y' = -x'$$

Word frequency (y) is inversely proportional to its rank (x).

Zipf's Law of Words in Different Languages



Zipf's Law of Population



Challenges in Extreme-scale Classification

- Data-sparse Challenge (labeled data are limited)
  - Remedy: Propagating model parameters over a graph of nodes (categories) if they are well connected (analogous to "borrow data" by the poor ones from the rich ones)
- Scalability Challenge (the sheer size of the label space)
  - Remedy: Divide and conquer via parallel computing and label clustering

13

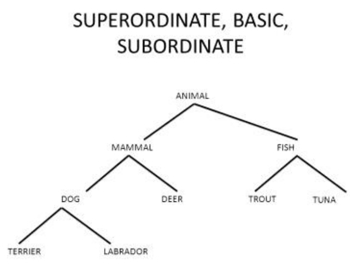
Outline

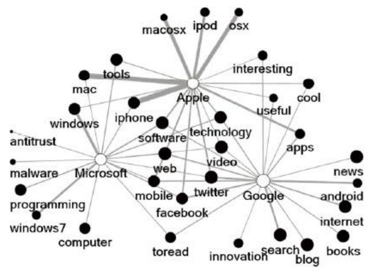
✓ Introduction to extreme-scale classification

- Leveraging label dependencies (hierarchical or graphical) in regularization [SIGKDD 2013]
- Neural models with label clustering for extreme-scale multi-label classification [NIPS 2019; SIGKDD 2020]

14

Two Types of Label Dependency Structures





Hierarchical

Graphical

Risk Minimization via Regularization

$$\hat{\mathbf{W}} = \arg\min_{\mathbf{W}} \{L_{emp}(\boldsymbol{W}, D_{train}) + C\phi(\boldsymbol{W})\}$$

Empirical Risk

Binary SVM : $\sum_{c}\sum_{i=1}^{N}(1 - y_i^{(c)}\boldsymbol{w}_c^T x_i)_+$

Binary LR: $\sum_{c}\sum_{i=1}^{N}\log(1 + \exp(-y_i^{(c)}\boldsymbol{w}_c^T x_i))$

Regularization Term

$$\phi(\boldsymbol{W}) = \sum_{c}\|\boldsymbol{w}_c\|^2$$

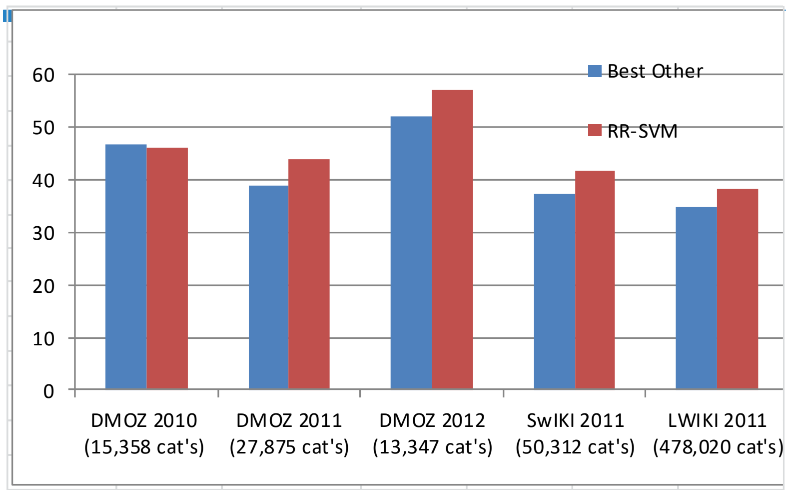
Ignoring the dependency structure among categories

Regularization with Structured Dependencies
(S Gopal & Y Yang, KDD 2013)

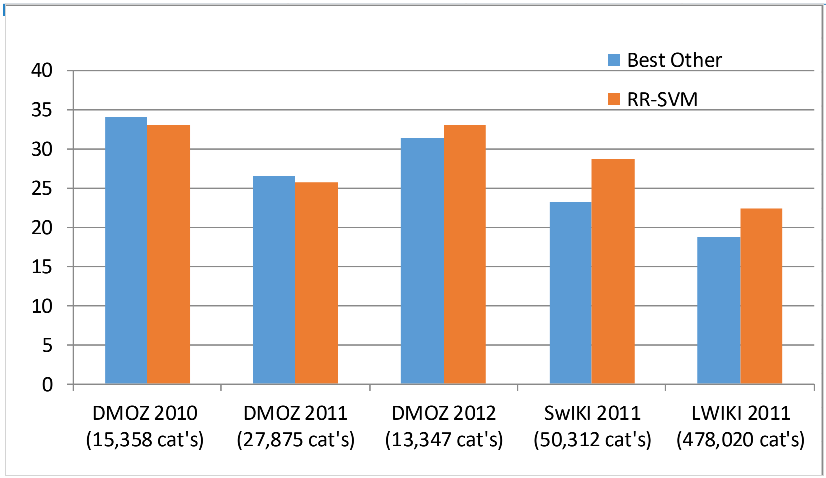
$$\widehat{\mathbf{W}} = \arg\min_{\mathbf{W}} \{\, L_{emp}(\boldsymbol{W}, D_{train}) + C\phi(\boldsymbol{W}) \,\}$$

- Given a hierarchy (H) of categories (nodes), we have
  $\phi_H(\mathbf{W}) = \sum_c ||w_c - w_{\pi(c)}||^2$
- Given a graph $G = (E, V)$ of categories (nodes), we have
  $\phi_G(\mathbf{W}) = \sum_{(i.j)\in E} ||w_i - w_j||^2$
- Iterative training w's on each node based on its parent node in H or the linked neighbors in G, until convergence.

Results of RR-SVM in *Micro*-avg F1 compared to other SOTA methods

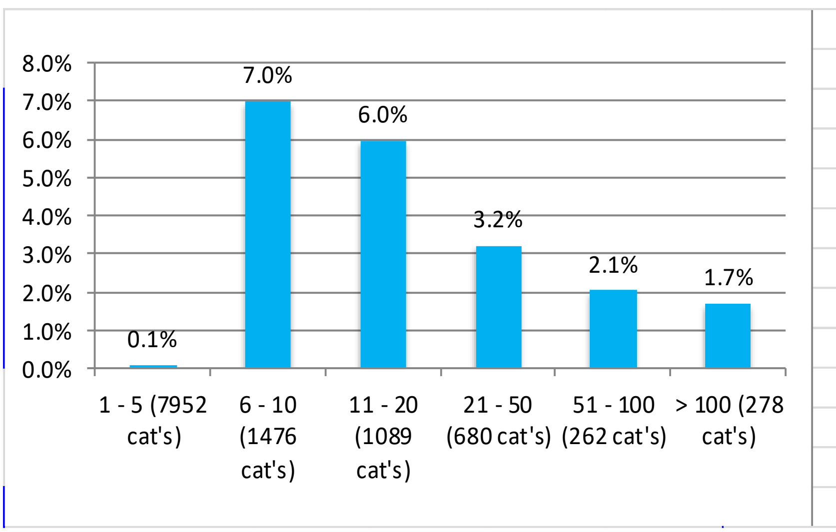


Results of RR-SVM in *Macro*-avg F1 compared to other SOTA methods



19

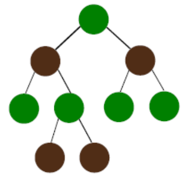
RR-SVM vs. BSVM on DMOZ-2012 (Macro-avg F1)



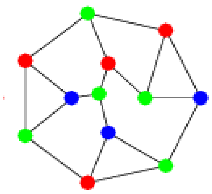
20

Divide-&-Conquer Strategies for Parallel Training

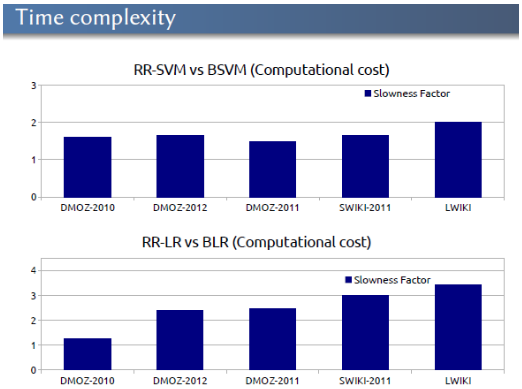
- **Hierarchies**
  - Optimize odd and even levels alternately



- **Graphs:** First find a graph vertex coloring, and then
  - Pick a color
  - In parallel, optimize all nodes with that color
  - Repeat with a different color



Training Time Comparison (at 2014)



On the LWIKI dataset
-- jointly optimizing 614,428 classifiers with 1 trillion parameters

BSVM	19 hours
RR-SVM	37 hours
BLR	36 hours
RR-LR	121 hours

using Hadoop with 500+ cores (300 cores as Mappers and 220 cores as reducers)

Outline

✓Introduction to extreme-scale classification

✓Leveraging label dependencies (hierarchical or graphical) in regularization [SIGKDD 2013]

- Neural models with label clustering for extreme-scale multi-label classification [NIPS 2019; SIGKDD 2020]

Xtransformer for XMC (W. Chang et al. KDD 2020)

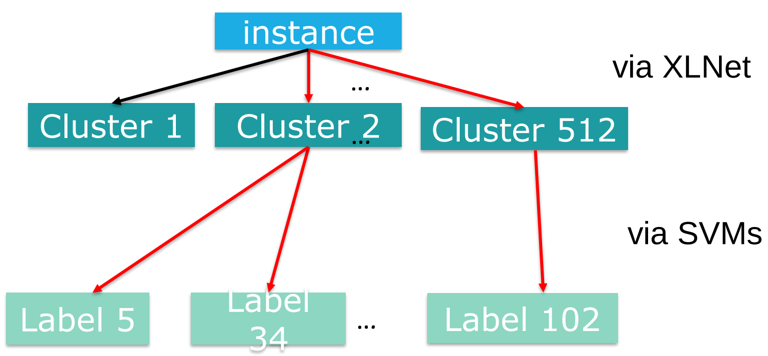
problem	XLNet-large model (# params) encoder	classifier	total	(batch size, sequence length)=(1,128) load model	+forward	+backward	+optimizer step
GLUE (MNLI)	361 M	2 K	361 M	2169 MB	2609 MB	3809 MB	6571 MB
XMC (1M)	361 M	1,025 M	1,386 M	6077 MB	6537 MB	OOM	OOM

- #categories (1M) and #parameters (1,385M) are extremely large
- Out-of-memory (OOM) for end-to-end training

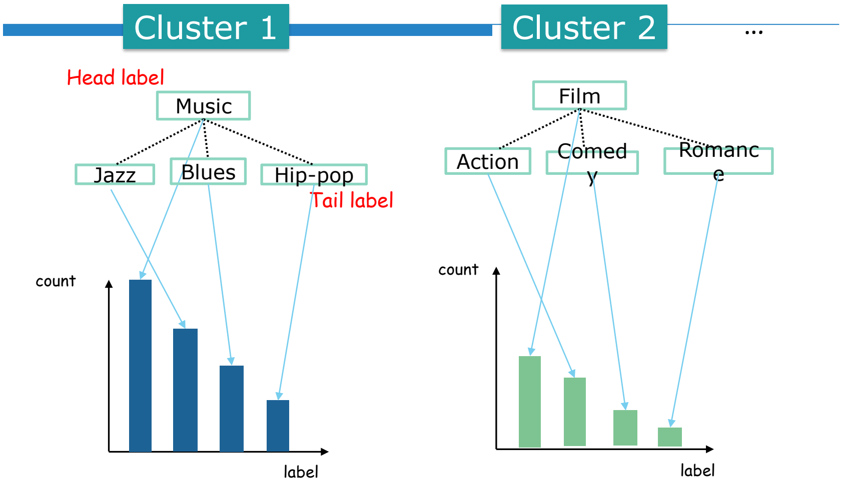
Clustering Labels for Divide & Conquer

- Create a vector representation for each label in **Wiki500k**
- Use **k-means clustering** to divide the labels into 512 clusters (with 1k labels per cluster)
- Train the system for a two-step classification of each test instance
  - Fine-tune **XLNet** (like Bert) for instance-to-cluster mapping (1-to-512 instead of 1-to-500k) as the first step
  - Train 1,000 SVM OVA (one vs. all) classifiers per cluster in parallel for the 2^{nd} step, i.e., within-cluster label prediction
  - Each 2^{nd} –level model is trained on a much smaller subset (only using the within-cluster instances)
  - **Why not using softmax for the 2^{nd} level?**

Two-step Classification Illustration



Label clustering based on semantic relevance



3 ways to obtain a vector representation (category profile) per label

- By summing up the word embeddings for each label name;
- By constructing a co-occurrence matrix (like in GloVe) of label pairs and applying a truncated SVD for a dimension-reduced vector per label; or
- By averaging of the embeddings of the words in the positive training instances (documents) of each label.

How to select negative instances for training each OVA classifier?

- Possible Strategies
  - ✓ For each target label, treat the training instances of all other labels as the negative ones – expensive (causing OOM in back-propagation)!
  - ✓ For each target label, treat the training instances of the other labels within the same cluster as the negative ones – better.
  - ✓ Treat the training instances of the system-predicted top-few clusters for additional negative ones to the above – even better.

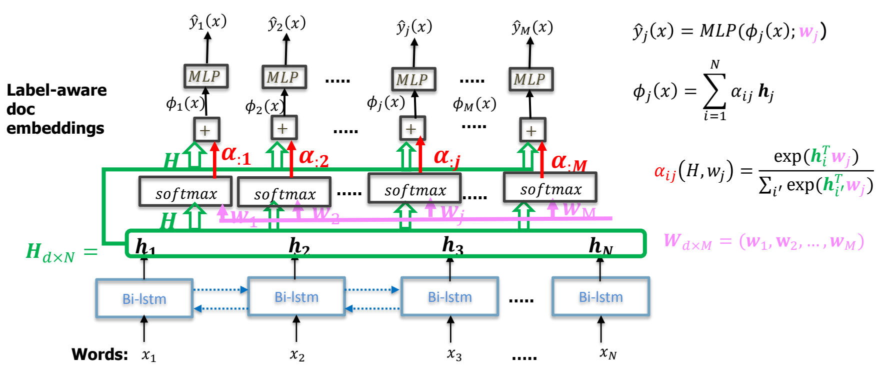
Representative Works in Extreme-scale Multi-label Classification

- SVM and LR models with recursive regularization over label hierarchies/graphs (S Gopal & Y Yang, KDD 2013)
- X-Transformer: Taming pre-trained Transformers (W. Chang et al. KDD 2020)
- AttentionXML: Label-tree based attention-aware deep model (Ronghui You et al. NeurIPS 2019)

AttentionXML: Label-tree based attention-aware deep model (Ronghui You et al. NeurIPS 2019)

- Key Ideas
  - Learning a label-aware document embedding per label for each document instead of one document embedding
  - Applying k-means recursively to generate a hierarchy of labels
  - Training one multi-label model at each lever of the hierarchy with down-sampling of negative training instances (for enhancing tractability)

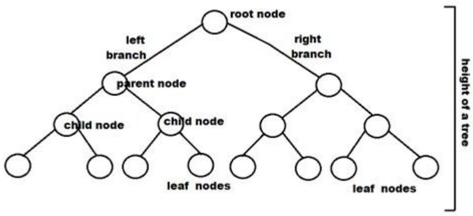
Labe-aware doc embedding
(for N documents and M categories)



Attention α_{ij} depends on both label parameter w_j and word embedding h_i .

Generating a hierarchy of labels
for large problem such as Wiki-500K

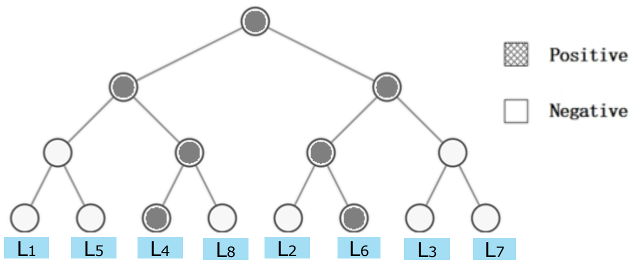
- Top-down application of K-means (K=2) to obtain a binary tree where the leaf nodes are category labels
- Collapse some levels to obtain a shallow balanced tree



$$\lceil \log_2 500{,}000 \rceil = 19$$

33

Node labeling over the hierarchy for each labeled training document

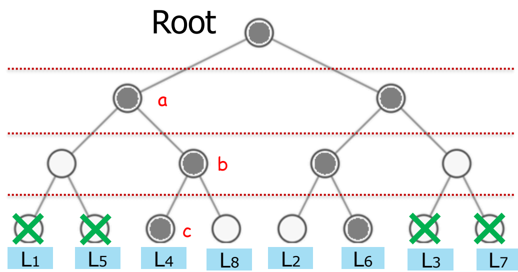


For example, if L4 and L6 are the correct labels for a document, then all the nodes along the paths from L4 and L6 to the root are also correct labels for this document.

34

Train a multi-label model per level

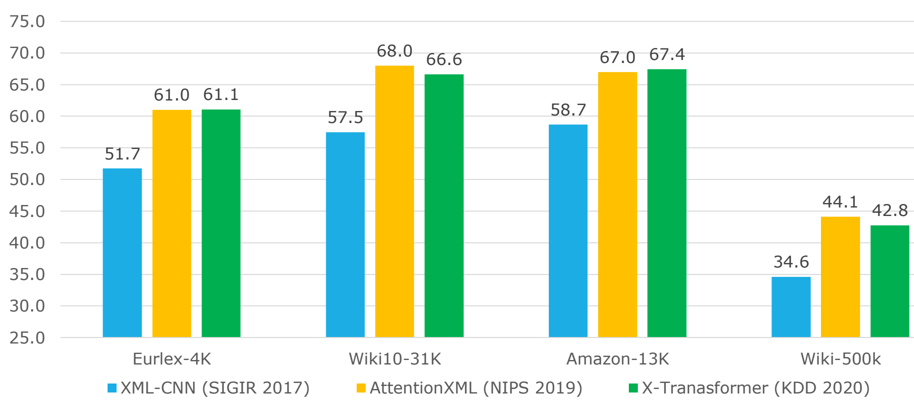
- In the training phase, the negative examples at each level only include those which share the same parents of a positive sibling.
- In the testing phase, the probability of leaf note L4 is estimated via the path from the root as $p_{L_4} = p_a p_b p_c$



35

Published results of SOTA methods

in P@5 on large benchmarks



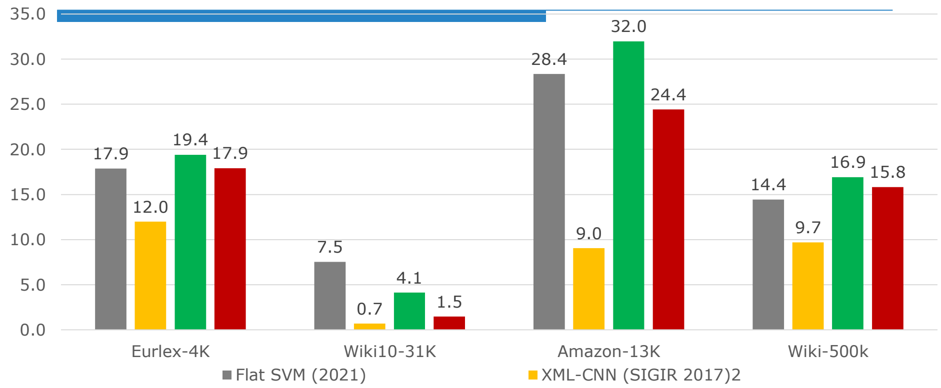
36

Issues with the published results

1) None of those methods have been evaluated in comparison with traditional methods (such as OVA SVMs or the recursively regularized SVM), so we cannot tell if the neural models work better.
2) The evaluation metric (P@5) is essentially a micro-averaging one, which does not necessarily reflect systems' performance on rare categories.
   - Should we use F1@5 instead in both micro-averaging and macro-averaging?
3) Why @5?
   - If the datasets have 5 labels per document, we can use F1@5.
   - ff the dataset has 19 labels per document, we should use F1@19.

SOTA Methods in Macro-avg F1@k on Large Benchmarks

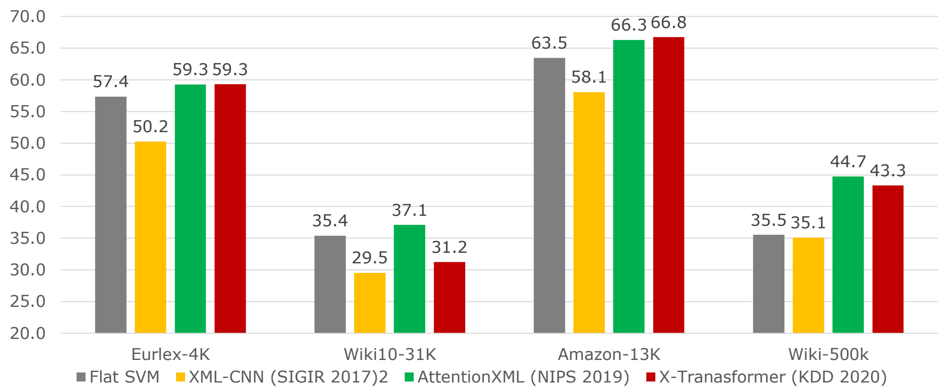
(k = 19 for Wiki10-30K and k = 5 for the other datasets)



Main Observations: SOTA neural models are not too much better than flat SVM models on average (we have not compared to RR-SVM yet.)

SOTA Methods in Micro-avg F1@k on Large Benchmarks

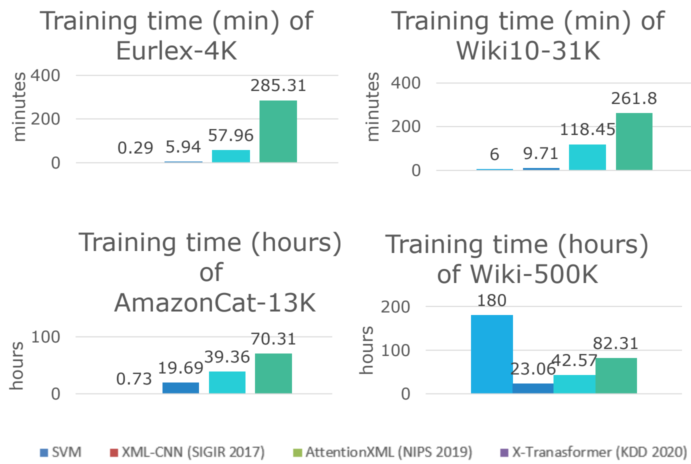
(k = 19 for Wiki10-30K and k = 5 for the other datasets)



Main Observations: SOTA neural models works better on large categories but not so on the tail (rare) categories.

39

Training Time: All the methods are trained on the same machine (SVM used 80-core CPU while the neural models used one-GPU 2080 TI), expect SVM on Wiki-500K used 5 servers, for which 180 h is the estimated time.



40

Concluding Remarks

- Extreme-scale classification is an important part of machine learning in the big-data era.
- Large category hierarchies/graphs present opportunities for structured learning.
- Neural learning has improved SOTA performance in some cases, probably due to contextualized representation learning. However, careful evaluation is needed for true insights.
- Data sparse issues remain open. How to leverage unlabeled data for few-shots learning is an active area for research.

References

- **Recursive regularization for large-scale classification with hierarchical and graphical dependencies** [pdf][slides][poster]
  Siddharth Gopal, Yiming Yang
  SIGKDD 2013 [Best student paper runner up]
- **AttentionXML: Label Tree-based Attention-Aware Deep Model for High-Performance Extreme Multi-Label Text Classification.** Ronghui You, Zihan Zhang, Ziye Wang, Suyang Dai, Hiroshi Mamitsuka, Shanfeng Zhu. NIPS 2019.
- **Deep Learning for Extreme Multi-label Text Classification** [web]
  Taming Pre-trained Transformers for eXtreme Multi-label Text Classification [pdf]
  Wei-Cheng Chang, Hsiang-Fu Yu, Kai Zhong, Yiming Yang, Inderjit S. Dhillon. SIG*KDD 2020*