Slice: Scalable Linear Extreme Classifiers trained on 100 Million Labels for Related Searches - Supplementary Material

Himanshu Jain*
himanshu.j689@gmail.com

Venkatesh Balasubramanian†
t-venkb@microsoft.com

Bhanu Chunduri†
bhanuc@microsoft.com

Manik Varma*†
manik@microsoft.com

ACM Reference Format:
Himanshu Jain, Venkatesh Balasubramanian, Bhanu Chunduri, and Manik Varma. 2019. Slice: Scalable Linear Extreme Classifiers trained on 100 Million Labels for Related Searches - Supplementary Material. In *The Twelfth ACM International Conference on Web Search and Data Mining (WSDM '19), February 11–15, 2019, Melbourne, VIC, Australia.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3289600.3290979

1 RELATED WORKS

Extreme multi-label learning: Extreme multi-label learning algorithms can be broadly categorised as either tree based [1, 13, 14, 24, 27], embedding based [4, 7, 8, 10, 17, 20, 30, 35] or 1-vs-All approaches [2, 18, 21, 23, 31–33]. Of these, 1-vs-All approaches are directly relevant to this paper.

1-vs-All approaches: 1-vs-All approaches such as DiSMEC [2], PD-Sparse [33], PPD-Sparse [32] and Parabel [23] learn a linear classifier per label. Many of these approaches usually have high prediction accuracies and low model sizes but suffer from high training and prediction times. For instance, DiSMEC [2] trains a label classifier on the entire training set leading to high training costs. PPD-Sparse [32] solves this problem to a limited extent by sampling the negative examples through a primal and dual sparsity preserving optimization procedure, thereby reducing its training time by upto 400x. However, in doing so, it looses on accuracy especially when the features are low dimensional and dense. Moreover, the speedup is not enough as it would still take days to train it on a dataset with 240 million data points and 100 million labels. In addition, 1-vs-All approaches, including DiSMEC, PDSparse, and PPDSparse, all require evaluating millions of classifiers per test point. This renders them infeasible for low latency and high throughput applications such as related searches. The recently proposed Parabel [23] reduces both training and prediction costs to logarithmic. It learns a balanced label hierarchy such that similar labels end up in the same leaf node. Negative training examples for a label are selected by taking positive examples of labels in the same leaf node. Unfortunately, Parabel's tree cannot be learnt accurately in low-dimensions as the linear separator learnt at each internal node does not have enough capacity to ensure that similar labels get partitioned together. Due to this, in case of low dimensional dense features, Parabel's prediction accuracy is significantly lower than that of DiSMEC.

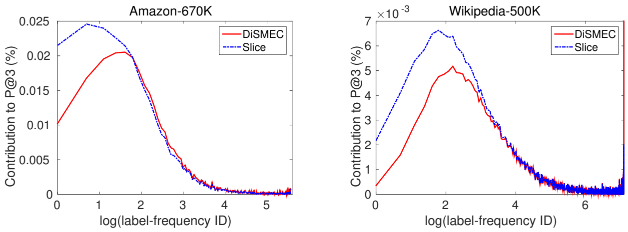
Deep extreme classification: Extreme classification methods have been shown to work well on high dimensional sparse bag-of-words features. Unfortunately, as stated before the performance of some of these methods, including Parabel [23] and PPD-Sparse [32], can be adversely affected when applied to low dimensional dense features. This becomes a cause of concern as the bag-of-words representation is inadequate for dealing with very short text and search engine queries. In such cases, low-dimensional deep features and word vector embeddings are more suitable. Approaches for learning such task-specific low-dimensional embeddings [18, 34] for extreme classification have recently been proposed. Unfortunately, these methods cannot be directly applied to the related searches problem as they are based on existing linear [18] or tree-based classifier [15] at the end of the deep network and thus cannot scale to multi-label problems with 100 million labels. This paper, therefore, focusses on the orthogonal problem of scaling Slice to 100 million labels using pre-trained embeddings such as C-DSSM [11, 26] or the embeddings generated by the deep learning for extreme multi-label text classification approach (XML-CNN) of [34].

Related searches: Most approaches for related searches can be broadly placed in three categories - sessions based [5, 6, 9, 22, 25, 28], query-url based [3, 19, 29] and those that generate synthetic queries [12, 16]. Sessions based approaches assume that the sequence of queries issued by a user within a short time interval are often closely related, as the user is trying to complete a search task, and thus can act as suggestions for each other. This co-occurrence information within a session is exploited by various algorithms in different ways. For instance, [5, 6] makes suggestions based on random walk scores on the query co-occurrence graph or approaches like [22] use learning to rank techniques to rank co-occurring queries. A common limitation of many sessions based algorithms [5, 6, 22, 25] is that they are limited to making suggestions for previously seen queries. Though this has been addressed by some deep learning based techniques [9, 28]. Unfortunately, they haven't been shown to scale to large problems. In contrast, Slice reformulates the problem as an extreme classification task and is therefore specifically designed for handling previously unseen queries. Query-url based approaches use query search results as a measure of similarity between two queries. Some [3] recommend suggestions that have search results similar to the input query, others [19] rank suggestions on the basis of hitting time on a bipartite

*Indian Institute of Technology Delhi
†Microsoft AI & Research

WSDM '19, February 11–15, 2019, Melbourne, VIC, Australia

ACM ISBN 978-1-4503-5940-5/19/02…$15.00
https://doi.org/10.1145/3289600.3290979



(a) (b)

Figure 1: Plot showing the contribution of each label frequency ($\sum_i y_{il}$) to the overall Precision@3. Slice is more accurate than DiSMEC on tail labels (smaller ID).

query-url graph. Finally, approaches such as [12, 16] generate synthetic queries by either dropping or by substituting some terms in the original query. Both these approaches tend to make suggestions that are similar to the input query. In contrast, Slice can make diverse suggestions that cover different aspects of the query. For example, for query "cam procedure shoulder" Slice is able to predict the suggestion "cost of arthroscopic shoulder surgery" as well as the suggestion "what to wear after shoulder surgery" which are relevant to the query and cover two completely different aspects of the query. Note that, some approaches [29] have been proposed that recommend *orthogonal* suggestions by filtering out queries having similar search results but often the suggestions aren't as diverse since their suggestion set is restricted to queries in the ad-hoc answers cache.

2 EXPERIMENTS

Evaluation metrics: Precision@k is defined as the average number of correct predictions among the top-k predictions of an algorithm while nDCG@k is just the weighted average, where the weight decreases logarithmically with rank . In particular, Precision@k and nDCG@k for a prediction $\hat{\mathbf{y}} \in \mathcal{R}^L$, given the ground truth label vector $\mathbf{y} \in \{0,1\}^L$, can be expressed as

$$\text{Precision@}k = \frac{1}{M}\sum_{i=1}^{M}\frac{1}{k}\left(\sum_{l\in \text{rank}_k(\hat{\mathbf{y}}_i)} y_{il}\right) \tag{1}$$

$$\text{nDCG@}k = \frac{1}{M}\sum_{i=1}^{M}\left(\frac{\sum_{l\in \text{rank}_k(\hat{\mathbf{y}}_i)} \frac{y_{il}}{\log(1+r_l)}}{\sum_{r=1}^{\min(k,\|\mathbf{y}_i\|_0)} \frac{1}{\log(1+r)}}\right) \tag{2}$$

where $\text{rank}_k(\hat{\mathbf{y}})$ returns the indices of the k largest elements of $\hat{\mathbf{y}}$ ranked in descending order and r_l is the corresponding rank.

Table 1: Results on Amazon-670K dataset with high dimensional sparse bag-of-words features.

Method	P1 (%)	P3 (%)	P5 (%)
Slice-s	44.35	39.63	36.20
DiSMEC	45.37	40.40	36.96
Parabel	44.90	39.81	35.99
PPDSparse	45.32	40.37	36.92
PfastreXML	39.46	35.81	33.05
SLEEC	35.05	31.25	28.56

Table 2: Results on publically available datasets with 100 dimensional GloVE embeddings. Results are similar to what were obtained using XML-CNN embeddings.

Dataset	Method	P1 (%)	P3 (%)	P5 (%)	N3 (%)	N5 (%)
EURLex-4K	Slice-s ($\|\mathcal{S}\| = 2000$)	**68.00**	**52.33**	**42.14**	**56.15**	**50.32**
	Slice-s	65.05	50.15	40.44	53.70	48.14
	Slice-Generative	30.35	24.28	20.14	25.75	23.54
	kNN-HNSW	63.30	49.52	40.38	52.83	47.70
	DiSMEC	67.93	52.35	42.06	56.13	50.24
	Parabel	63.17	48.38	39.31	51.98	46.84
	PPDSparse	66.50	50.85	40.60	54.64	48.71
	PfastreXML	63.80	49.09	40.10	52.60	47.53
Amazon-670K	Slice-s ($\|\mathcal{S}\| = 2000$)	**32.90**	**29.07**	**25.87**	**30.91**	**29.38**
	Slice-s	28.19	25.43	22.95	26.95	25.91
	Slice-Generative	23.81	20.50	17.87	22.02	20.68
	kNN-HNSW	20.19	18.09	16.77	19.13	18.62
	DiSMEC	32.08	28.31	25.36	30.00	28.56
	Parabel	25.13	21.66	19.14	23.14	21.84
	PPDSparse	18.35	15.90	14.03	16.89	15.89
	PfastreXML	29.47	26.74	24.83	28.29	27.61
Wikipedia-500K	Slice-s ($\|\mathcal{S}\| = 2000$)	43.03	28.83	22.19	36.49	35.48
	Slice-s	39.76	26.67	20.55	33.62	32.80
	Slice-Generative	26.02	16.89	13.03	21.93	21.65
	kNN-HNSW	43.16	27.53	20.66	34.78	32.93
	DiSMEC	**44.10**	**29.94**	**23.14**	**37.42**	**36.21**
	Parabel	40.61	26.44	19.92	33.22	31.51
	PPDSparse	25.39	17.32	13.91	21.41	21.11
	PfastreXML	39.73	25.68	19.66	32.45	31.06

REFERENCES

[1] R. Agrawal, A. Gupta, Y. Prabhu, and M. Varma. 2013. Multi-label Learning with Millions of Labels: Recommending Advertiser Bid Phrases for Web Pages. In *WWW*.

[2] R. Babbar and B. Shoelkopf. 2017. DiSMEC-Distributed Sparse Machines for Extreme Multi-label Classification. In *WSDM*.

[3] R. Baeza-Yates, C. Hurtado, and M. Mendoza. 2004. Query recommendation using query logs in search engines. In *ICDT*.

[4] K. Bhatia, H. Jain, P. Kar, M. Varma, and P. Jain. 2015. Sparse Local Embeddings for Extreme Multi-label Classification. In *NIPS*.

[5] P. Boldi, F. Bonchi, C. Castillo, D. Donato, A. Gionis, and S. Vigna. 2008. The query-flow graph: model and applications. In *CIKM*.

[6] F. Bonchi, R. Perego, F. Silvestri, H. Vahabi, and R. Venturini. 2012. Efficient query recommendations in the long tail via center-piece subgraphs. In *SIGIR*.

[7] Y. N. Chen and H. T. Lin. 2012. Feature-aware Label Space Dimension Reduction for Multi-label Classification. In *NIPS*.

[8] M. Cissé, N. Usunier, T. Artières, and P. Gallinari. 2013. Robust Bloom Filters for Large MultiLabel Classification Tasks. In *NIPS*.

[9] M. Dehghani, S. Rothe, E. Alfonseca, and P. Fleury. 2017. Learning to attend, copy, and generate for session-based query suggestion. In *CIKM*.

[10] D. Hsu, S. Kakade, J. Langford, and T. Zhang. 2009. Multi-Label Prediction via Compressed Sensing. In *NIPS*.

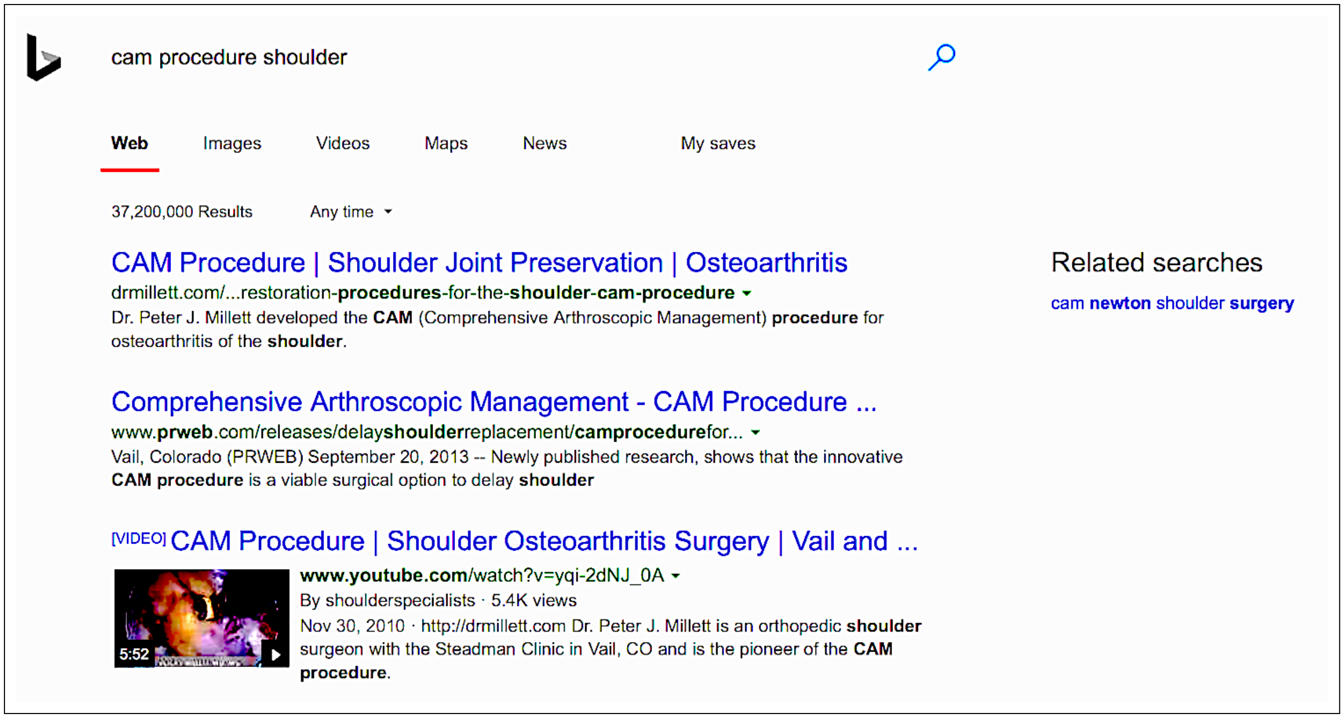
[11] P. S. Huang, X. He, J. Gao, L. Deng, A. Acero, and L. Heck. 2013. Learning deep structured semantic models for web search using clickthrough data. In *CIKM*.

[12] A. Jain, U. Ozertem, and E. Velipasaoglu. 2011. Synthesizing high utility suggestions for rare web search queries. In *SIGIR*.

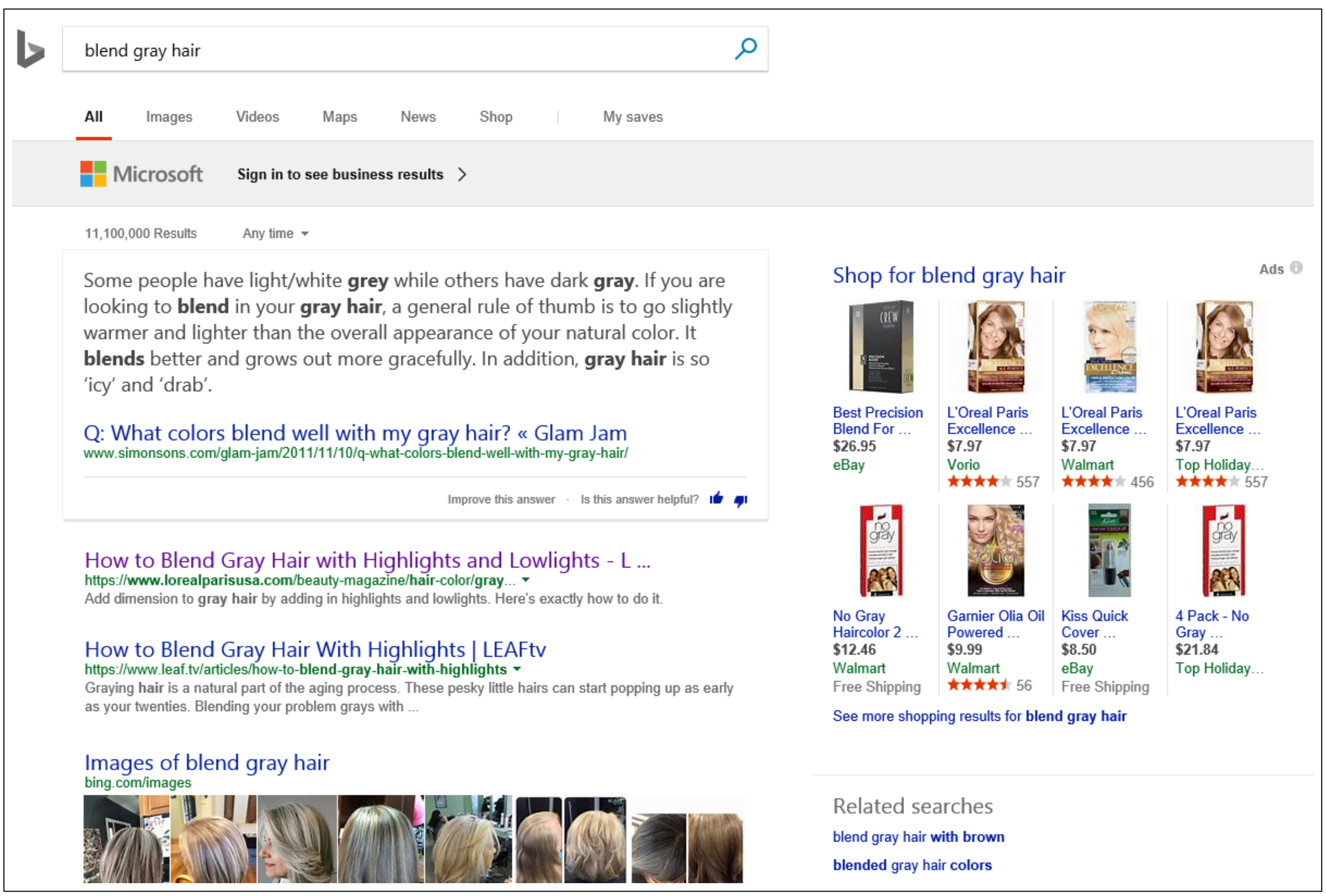
[13] H. Jain, Y. Prabhu, and M. Varma. 2016. Extreme Multi-label Loss Functions for Recommendation, Tagging, Ranking & Other Missing Label Applications. In *KDD*.

[14] K. Jasinska, K. Dembczynski, R. Busa-Fekete, K. Pfannschmidt, T. Klerx, and E. Hüllermeier. 2016. Extreme F-measure Maximization Using Sparse Probability Estimates. In *ICML*. 1435–1444.

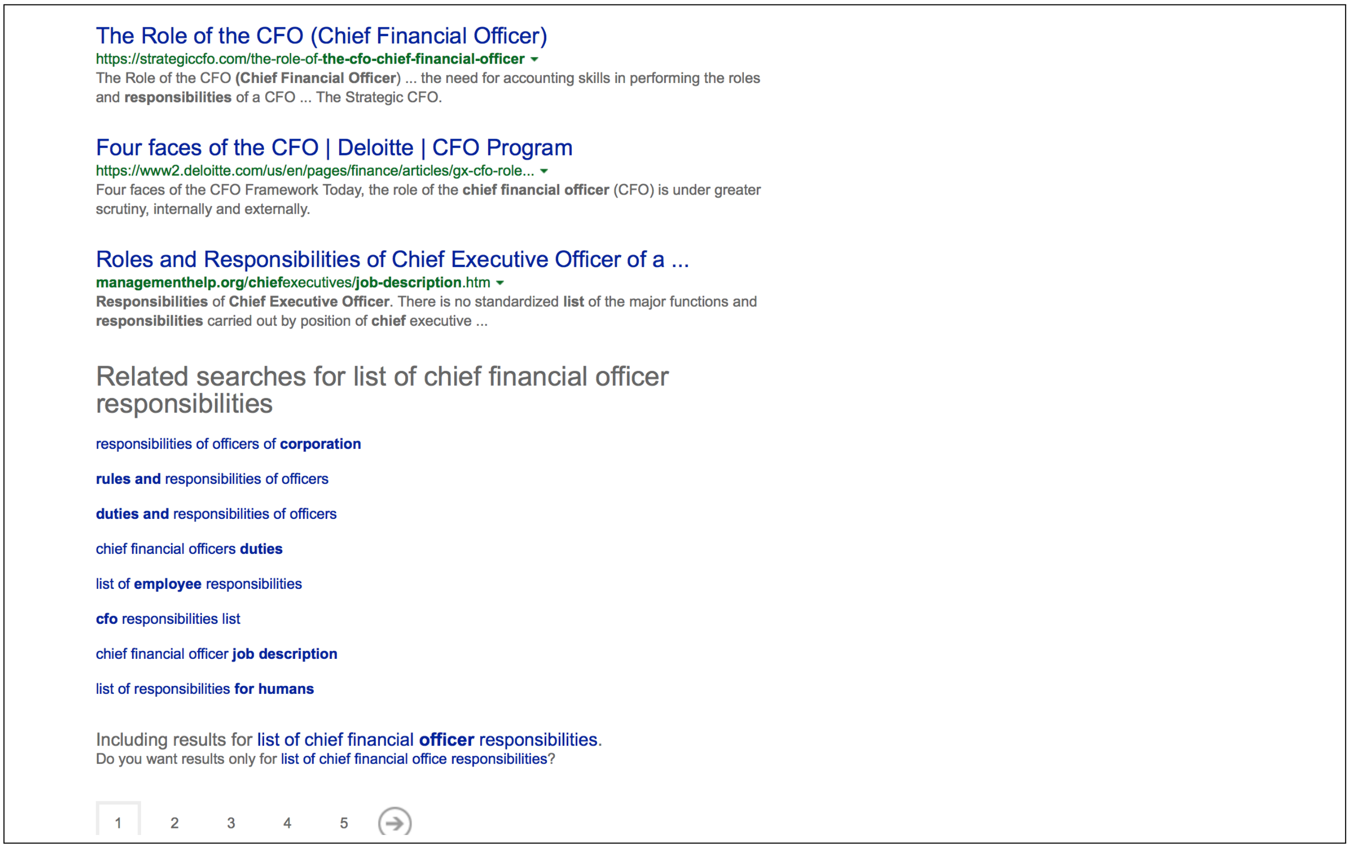
Figure 2: Screenshots of Bing related searches results for various queries.



(a)



(b)



(c)

[15] Y. Jernite, A. Choromanska, and D. Sontag. 2017. Simultaneous Learning of Trees and Representations for Extreme Classification and Density Estimation. In *ICML*.
[16] R. Jones, B. Rey, O. Madani, and W. Greiner. 2006. Generating query substitutions. In *WWW*.
[17] Z. Lin, G. Ding, M. Hu, and J. Wang. 2014. Multi-label Classification via Feature-aware Implicit Label Space Encoding. In *ICML*.
[18] J. Liu, W. C. Chang, Y. Wu, and Y. Yang. 2017. Deep Learning for Extreme Multi-label Text Classification. In *SIGIR*.
[19] Q. Mei, D. Zhou, and K. Church. 2008. Query Suggestion Using Hitting Time. In *CIKM*.
[20] P. Mineiro and N. Karampatziakis. 2015. Fast Label Embeddings for Extremely Large Output Spaces. In *ECML*.
[21] A. Niculescu-Mizil and E. Abbasnejad. 2017. Label Filters for Large Scale Multilabel Classification. In *AISTATS*.
[22] U. Ozertem, O. Chapelle, P. Donmez, and E. Velipasaoglu. 2012. Learning to suggest: a machine learning framework for ranking query suggestions. In *SIGIR*.
[23] Y. Prabhu, A. Kag, S. Gopinath, K. Dahia, S. Harsola, R. Agrawal, and M. Varma. 2018. Extreme multi-label learning with label features for warm-start tagging, ranking and recommendation. In *WSDM*.
[24] Y. Prabhu and M. Varma. 2014. FastXML: A fast, accurate and stable tree-classifier for extreme multi-label learning. In *KDD*.
[25] E. Sadikov, J. Madhavan, L. Wang, and A. Halevy. 2010. Clustering query refinements by user intent. In *WWW*.
[26] Y. Shen, X. He, J. Gao, L. Deng, and G. Mesnil. 2014. Learning semantic representations using convolutional neural networks for web search. In *WWW*.
[27] S. Si, H. Zhang, S. S. Keerthi, D. Mahajan, I. S. Dhillon, and C. J. Hsieh. 2017. Gradient Boosted Decision Trees for High Dimensional Sparse Output. In *ICML*. 3182–3190.
[28] A. Sordoni, Y. Bengio, H. Vahabi, C. Lioma, J. Grue Simonsen, and J. Y. Nie. 2015. A hierarchical recurrent encoder-decoder for generative context-aware query suggestion. In *CIKM*.
[29] H. Vahabi, M. Ackerman, D. Loker, R. Baeza-Yates, and A. Lopez-Ortiz. 2013. Orthogonal query recommendation. In *RecSys*.
[30] J. Weston, S. Bengio, and N. Usunier. 2011. Wsabie: Scaling Up To Large Vocabulary Image Annotation. In *IJCAI*.
[31] J. Weston, A. Makadia, and H. Yee. 2013. Label Partitioning For Sublinear Ranking. In *ICML*.
[32] I. E. H. Yen, X. Huang, W. Dai, P. Ravikumar, I. Dhillon, and E. Xing. 2017. PPDsparse: A Parallel Primal-Dual Sparse Method for Extreme Classification. In *KDD*. 545–553.
[33] I. E. H. Yen, X. Huang, P. Ravikumar, K. Zhong, and I. S. Dhillon. 2016. PD-Sparse: A primal and dual sparse approach to extreme multiclass and multilabel classification. In *ICML*.
[34] W. Zhang, L. Wang, J. Yan, X. Wang, and H. Zha. 2017. Deep Extreme Multi-label Learning. *CoRR* (2017).
[35] Y. Zhang and J. G. Schneider. 2011. Multi-Label Output Codes using Canonical Correlation Analysis. In *AISTATS*.

Table 3: Results on extreme classification datasets in terms of propensity scored precision@k.

Method	PSP1 (%)	PSP3 (%)	PSP5 (%)
EURLex-4K			
Slice-s	33.17	39.84	**43.08**
Slice-Generative	**38.71**	**40.39**	41.19
DiSMEC	29.52	36.09	39.45
Parabel	30.03	36.37	39.76
PfastreXML	36.80	39.65	41.65
SLEEC	28.77	34.28	37.60
Amazon-670K			
Slice-s	24.48	**27.01**	**29.07**
Slice-Generative	**25.85**	26.67	27.50
kNN-HNSW	17.21	19.79	22.11
DiSMEC	21.40	24.79	27.77
Parabel	19.45	21.93	23.66
PPDSparse	16.85	19.60	22.08
PfastreXML	20.36	21.01	21.81
SLEEC	6.98	8.31	9.53
Wikipedia-500K			
Slice-s	27.51	**29.82**	**32.00**
Slice-Generative	**29.02**	28.08	29.30
kNN-HNSW	22.62	25.19	27.26
DiSMEC	23.28	26.64	29.16
Parabel	21.03	24.28	26.70
PfastreXML	25.79	24.96	25.93

Table 4: Prediction accuracy of related searches algorithms becomes worse when their suggestion set is not restricted to set on which Slice was trained.

Dataset	Method	P1 (%)	P3 (%)	P5 (%)	N3 (%)	N5 (%)
RS-2M	M1	32.71	25.54	21.46	30.89	30.73
	M2	38.15	27.35	21.87	38.16	29.35
	M3	6.70	5.50	5.30	6.31	6.78
	M4	23.84	18.43	15.20	21.59	20.86
	M4	13.79	10.23	8.25	11.90	11.20
	M6	2.60	2.70	3.00	2.80	3.38
	Slice-s	**43.08**	**31.10**	**24.81**	**45.40**	**47.60**
RS-33M	M1	29.22	21.64	17.53	27.35	27.12
	M2	25.44	21.07	18.06	26.99	28.83
	M3	7.86	6.53	6.06	7.76	8.45
	M4	23.52	18.47	15.34	22.66	22.92
	M5	15.24	11.06	8.80	13.65	13.34
	M6	4.58	4.50	4.73	5.13	6.24
	Slice-s	**39.21**	**29.67**	**24.15**	**39.08**	**40.19**

Table 5: Related searches recommendations by Bing and Slice. Bing recommended less than three suggestions for tail queries in (a) and (b) and recommended poor suggestions for input query in (c) while Slice provided eight relevant and diverse suggestions for all the input queries.

Bing	Slice
(a) cam procedure shoulder	
cam newton shoulder surgery	shoulder replacement surgery success rate
	types of shoulder surgery procedures
	cost of arthroscopic shoulder surgery
	alternative to shoulder replacement surgery
	what to expect after shoulder arthroscopy
	shoulder replacement side effects
	total shoulder replacement video
	shoulder surgery procedures
(b) blend gray hair	
blend gray hair with brown	blending gray hair with lowlights
blended gray hair colors	blend gray hair with brown
	hair color that blends with gray
	blending gray hair instead of covering it
	how to go gray without looking old
	how to enhance grey hair
	reverse gray hair naturally
	silver highlights to blend gray
(c) list of chief financial officer responsibilities	
responsibilities of officers of corporation	chief financial officer job description
rules and responsibilities of officers	what is a chief financial officer
duties and responsibilities of officers	chief financial officer salary
chief financial officers duties	chief financial officer qualifications
list of employee responsibilities	chief financial officer requirements
cfo responsibilities list	chief financial officer jobs
chief financial officer job description	finance officer duties and responsibilities
list of responsibilities for humans	cfo duties for small company