Search Filter Ranking with Language-Aware Label Embeddings

Jacek Golebiowski*
jacekgo@amazon.com
Amazon Search
Germany

Felice Antonio Merra*†
felice.merra@poliba.it
Politecnico di Bari
Italy

Ziawasch Abedjan
ziawasch@amazon.com
Amazon Search, Leibniz University Hannover, L3S Research Center
Germany

Felix Biessmann‡
biessman@amazon.com
Berlin University of Applied Sciences, Einstein Center Digital Future
Germany

ABSTRACT

A search on the major eCommerce platforms returns up to thousands of relevant products making it impossible for an average customer to audit all the results. Browsing the list of relevant items can be simplified using search filters for specific requirements (e.g., shoes of the wrong size). The complete list of available filters is often overwhelming and hard to visualize. Thus, successful user interfaces desire to display only the ones relevant to customer queries.

In this work, we frame the filter selection task as an extreme multi-label classification (XMLC) problem based on historical interaction with eCommerce sites. We learn from customers' clicks and purchases which subset of filters is most relevant to their queries treating the relevant/not-relevant signal as binary labels.

A common problem in classification settings with a large number of classes is that some classes are underrepresented. These rare categories are difficult to predict. Building on previous work we show that classification performance for rare classes can be improved by accounting for the language structure of the class labels. Furthermore, our results demonstrate that including language structure in category names enables relatively simple deep learning models to achieve better predictive performance than transformer networks with much higher capacity.

CCS CONCEPTS

• **Information Retrieval** → **Ranking**; *Search Filters*; • **Neural Networks** → Joint Input-Output embedding.

*Both authors contributed equally to this research.
†Work done while at Amazon Search.
‡Work done while at Amazon Search.

WWW '22 Companion, April 25–29, 2022, Virtual Event, Lyon, France

ACM ISBN 978-1-4503-9130-6/22/04.
https://doi.org/10.1145/3487553.3524218

KEYWORDS

Information Retrieval, Ranking, Search Filters

ACM Reference Format:
Jacek Golebiowski, Felice Antonio Merra, Ziawasch Abedjan, and Felix Biessmann. 2022. Search Filter Ranking with Language-Aware Label Embeddings. In *Companion Proceedings of the Web Conference 2022 (WWW '22 Companion), April 25–29, 2022, Virtual Event, Lyon, France.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3487553.3524218

1 INTRODUCTION

A standard search in an eCommerce store, e.g., Amazon or eBay, returns thousands of products, making it difficult for the customer to find the set of items that they might like. Product discovery can be facilitated through search filters used to restrict the set of items based on a specific attribute value, e.g., a t-shirt of a specified color or TV of a specified size. However, the large set of filters available in most online retail stores is impossible to navigate by customers making it necessary to identify a smaller subset of the most relevant filters per query. Finding the most relevant subset of labels, i.e., search filters, given textual input, i.e., a search query, is a common problem that could be treated both as a recommendation [15] or a multi-label classification task. Recent work has shown that optimizing for the classification objective can lead to highly accurate models [8] and this approach is the focus of this work.

Most ranking and classification datasets with many labels, such as the one considered in this work, exhibit a power-law distribution over label frequencies. The sparse representation of labels as unique identifiers (IDs) can make it challenging to learn from and about the uncommon targets. A common strategy to learn about tail labels is to integrate side information associated with them. This technique has been demonstrated to achieve high accuracy for large-scale classifiers [8]. Such side information is often available in industrial datasets where classification targets correspond to real-world items (here, search filters). Open datasets also contain corresponding descriptions, for example, the Wiki family dataset [30], Amazon family [14], and EuroLex [13] include label names that harbor semantic information. Yet, conventional document-based XMLC models treat labels as atomic symbols, ignoring extra

information. While some methods in the literature leverage label names for data pre-processing [7] or for dealing with misspelled categorical string labels [6], the information in labels is rarely used as first-class inputs in text classification models.

Motivated by those observations in real-world eCommerce platforms, we focus on improving the performance of XMLC models on the infrequent labels by exploiting the signals extractable from a label's textual description. Our contributions are summarized as follows: (i) we perform an experimental analysis of seven state-of-the-art (SOTA) XMLC models and one tag recommender system on a query-to-search-filters dataset collected from a large eCommerce platform; (ii) we evaluate the accuracy of different neural architectures that make use of the label names. We verify the performance of the tested models on several evaluation metrics to assess their accuracy on infrequent labels and analyze their shortcomings; and finally (iii) we propose Language-Aware Label Embeddings (LALE) that allows to integrate the language structure of category names into the learning objective.

2 RELATED WORK

Predicting the relevance of filters for a a query can be formalized as a ranking or an extreme multi-label classification task. In the first interpretation, a tag recommender system (TRS) builds a ranked list of tags for each input. A standard method such as WSABIE [22] implements a factorization-based recommender with a WARP loss. Over the years, various models have extended WSABIE by integrating features extracted by neural models [11, 28, 28] further improving accuracy.

In the XMLC formulation, the goal is to directly predict the probability of a filter/label having given a query. There are three main classes of models used in XMLC, embedding-based, tree-based, and deep learning. *Embedding-based* models use factorization approaches to learn the similarity between the query-label embeddings. AnneXML [21] is one of the SOTA models from this class. They are not without limitations, for instance, SLEEC [4] shows difficulties in recommending long-tail labels due to the inability to project million-sized labels into a tiny latent space. *Tree-based* models learn a hierarchical tree structure within the labels, grouping similar ones together. Standard models are: FastXML [18] that is optimized by an nDCG-based ranking loss function, PfastreXML [18], which improves FastXML integrating propensity scored losses to promote the prediction of infrequent but rewarding tail labels, and, Parabel [17], which recursively partitions the labels into two balanced groups. *Deep Learning-based* models, extracting the dense textual representation from input documents [1, 29], were spearheaded by XML-CNN [12]. Following that, AttentionXML [26] (after XBERT [5]) was developed to capture the sequential information of the text by the Bi-LSTM model and the Attention mechanism. APLC-XLNet [25] extended XLNet [24] building an Adaptive Probabilistic Label Tree to approximate the cross-entropy loss by exploiting the unbalanced label distribution to form clusters that explicitly reduce the computational time. Recently, X-Transformer [7] has been proposed, which fine-tunes BERT models on clusters of labels and trains a linear ranking model to recommend the labels into a cluster of labels. This work has been continued resulting in the PECOS model [27]. Finally, GILE - a text-classification model that integrates label textual features based on the idea that joint input-label text embeddings overcome the generalization limits on unseen or long-tail labels [16].

3 METHODOLOGY

Motivated by the eCommerce scenario, we propose a suite of models to rank search filters for a given user query. Let $q \in Q$ be a query in the set of queries Q and let $f \in F$ be a filter in the set F of the available filters. We define the **query to filter**(Q2F) task as a multi-classification problem such that

$$\forall q \in \mathcal{Q}, f'_q = \arg\max_{f \in \mathcal{F}} g\left(q, f\right)$$

where f'_q is a filter suggested when typing the query q, and $g : \mathcal{Q} \times \mathcal{F} \rightarrow \mathbb{R}$ is the utility function that has to be maximized such that the recommended filter is selected when the q is typed.

To this end, we investigate a four different formulations of the function g, including the LALE model. We first introduce the representations for customer queries and search filters and following that, we discuss how those signals are combined to obtain the query-filter scores.

Query Representation The first component of our proposal is the query encoder. The input embedding module converts a list of input subword tokens into a list representation via the subword lookup table, followed by a single Bi-LSTM layer and a fully connected layer afterward to obtain the vector representation of the query.

Concretely, we model each query as a sequence of S subwords tokens (end-padded with zeros) extracted from a 32k-dimensional dictionary of subword units. Subwords are found via byte pair encoding [19] applied on the queries found in the training set and label names. Each subword is encoded as a learnable, G dimensional vector making the query encoding a $S \times G$ matrix. Each subword-encoded query is given as input to a BiLSTM layer network to learn a latent representation; the network uses bias terms and the hyperbolic tangent function as activation. We only use the final output of the network (concatenation of final states from both directional BiLSTM) to compute the final embedding given as a fixed size feature vector $q \in \mathbb{R}^H$. The BiLSTM output q is passed through a fully connected layer with relu activations to get the E dimensional, query representation $\hat{q}$. The architecture was chosen by testing the BiLSTM-based, CNN-based, and BOW-based encoders with the same number of parameters w.r.t. precision@1 on the validation set.

Filter Representation. The second main element of the architecture is the filters (label) encoder. We design an encoder that considers two types of label information: the atomic filter identifiers and the filter textual description (label names). Filter names are embedded with the same architecture use

for processing queries (subword lookup, Bi-LSTM, and a fully connected layer), and the filter IDs are embedded using a simple lookup table with one row per unique ID. The two embedding models (query text and filter text) share the subword lookup table but do not share any other parameters. In the following sections, we will refer to the filter names embeddings as $\hat{f}_t$

To embed the IDs, each filter f in the set F is embedded as a Z-dimensional vector and denoted as $\hat{f}_{id}$. The embeddings are taken from a $|F| \times Z$ matrix of embeddings where i-th row corresponds to the embedding of a filter with ID=i and Z is a hyper-parameter which needs to be equal to E.

Baselines. Based on the presented architecture, we construct three variants of the baseline classifier: the combined, text-only and ID-only baselines where filters are embedded using text and IDs, text-only and IDs-only respectively; the three options are described below. Two separate types of filter representations allow us to build three different embeddings denoted as $\widetilde{f}$:

$$\widetilde{f} = \begin{cases} \hat{f}_{id}, & \text{if } \textbf{ID}\text{-only} \\ \hat{f}_t, & \text{if } \textbf{Text}\text{-only} \\ tanh\left(W'_e \cdot concat\left(\hat{f}_{id}, \hat{f}_t\right)\ B\right) & \text{if } \textbf{Combined} \end{cases} \quad (1)$$

where, W'_e is the matrix embedding of a fully-connected layer used in the **Combined** scenario to project the concatenation of the id-based and text-based embedding to the E dimensional space of the query embedding $\hat{q}$ and B is the bias term.

Given a query embedding $\hat{q}$ and a label embedding $\widetilde{f}$, we measure a recommendation (classification) score $g\left(q, f\right) = \hat{s}_{q,f}$ with a dot product between the $\left(q, f\right)$-latent representations $\hat{s}_{q,f} = \hat{q} \cdot \widetilde{f}$.

LALE. We propose the Language-Aware Label Embedding (LALE) architecture, inspired by the GILE model [16]. LALE integrates the textual and unique (ID-based) representation of labels by producing a joint ID-text representation of each filter $\widetilde{f'}$ using element-wise multiplication. Once the filter representation is computed, the query-filter affinity score $g\left(q, f\right) = \hat{s}_{q,f}$ can be found by taking a dot product between $\hat{q}$ and $\widetilde{f'}$ as before.

Formally, the label relevance score is given as the similarity between the embeddings of the query and each filter as $\hat{s}_{q,r} = \hat{q} \cdot \left(\hat{f}_{id} \circ \hat{f}_t\right)$ where, $\hat{f}_{id}$, $\hat{f}_t$ and $\hat{q}$ are defined as before and $\circ$ is an element-wise multiplication. We hypothesize that this formulation allows the model to learn the base representation of each filter based on text (similar for filters with similar names) and an ID-dependent perturbation for each unique label. What is more, this method of combining all three input signals enforces a joint embedding of all representations in a common space.

This formulation could be also re-written in the form of $\hat{s}_{q,f} = \hat{f}_{id} \cdot \left(\hat{q} \circ \hat{f}_t\right)$ where the textual representation of labels and query are combined using element-wise multiplication and the output is combined with the filter ID embeddings using an inner product to get the classification scores. In this view, the filter ID embeddings D_f can be thought of as weights of a fully connected layer processing $\hat{q} \circ \hat{f}_t$. This interpretation is similar to a vector of global parameters for the final dense layer as proposed in [16]. However, we use a unique set of parameters for each label to encode the necessary, filter-specific information.

Model Training To learn the parameters of the proposed models, we optimize the point-wise binary cross-entropy based on model output passed through a sigmoid function. We use the Adam [10] algorithm for optimisation.

4 EXPERIMENTS

Dataset. We conduct the experiments on a real-world dataset sampled from the logs of a large e-commerce site with search queries (as input), IDs of selected search filters (as labels), and the names of those search filters (as label names). The dataset does not include any personally identifying information about the users who performed the searches. The dataset contains 1,573,137 searches with 603,217 unique search keywords and 1600 unique labels. It has been divided into train, validation, and test set with the proportion 8:1:1 with no duplicate queries between the three datasets. We process both validation and test set by merging the duplicated queries and concatenating all filters active in the duplicates. The distribution of labels appearing in our dataset follows a power law, and the most common labels are seen significantly more often than the uncommon ones. To address the two label regimes, we have split the labels into two sets: the most popular 200, which account for 87% of all label occurrences, are known as the head filters, and the remainder 1400 are denoted as tail labels.

Evaluation. We compare our solutions against XMLC and TRS benchmark models with classical accuracy metrics, i.e., precision and normalized discounted cumulative gain (nDCG); we do not show precision@1 since it is equal to ndcg@1. Results, found in Table 1, are an average over ten runs with random initialization; the values are more than one standard deviation apart unless stated otherwise. We follow the conventional evaluation with beyond-accuracy metrics: F1 score on different label classes (head and tail) with equal weighting across classes, and search filters (labels) coverage (fraction of labels that were ever predicted as relevant).

Benchmark Models We use selected state of the art classification and ranking models to benchmark our presented methods. Concretely, Parabel [17], Bonsai [9], AnneXML [21], PfastreXML [18], FastXML [18], APLC-XLNet [25] (denoted as APLC) and the GILE [16] model as XMLC baselines and the TagSpace [23] method for comparison with ranking. All methods used for benchmarking are discussed in Section 2.

Reproducibility When training proposed baselines described in Section 3, we use G=256 dimensional subword embeddings, H=256 dimensional Bi-LSTM network followed by a E=256 dimensional fully connected layers. The label ID and text embedding are combined using a Fully connected layer with F=256 outputs. The label ID embedding relies on Z=256 dimensional vectors. The hyper parameters of the model were found using grid-based HPO on the validation dataset

Table 1: Accuracy Results with best values in bold.

Model	ndcg@1	ndcg@3	ndcg@5	p@3	p@5
Parabel	0.5055	0.5996	0.6432	0.3064	0.2246
Bonsai	0.4017	0.4416	0.4555	0.2207	0.1506
AnnexXML	0.4859	0.5779	0.6233	0.2988	0.2212
PfastreXML	0.4999	0.5998	0.6468	0.3079	0.2279
FastXML	0.5147	0.6103	0.6556	0.3106	0.2288
APLC	0.3506	0.4101	0.4382	0.2083	0.1516
Gile	0.2070	0.3640	0.4010	0.2188	0.1649
TagSpace	0.4186	0.5314	0.5903	0.2776	0.2156
ID-only	0.5193	0.6087	0.6554	0.3150	0.2342
Text-only	0.5022	0.5762	0.6070	0.2931	0.2078
Combined	0.5201	0.6096	0.6559	0.3155	0.2344
LALE (ours)	**0.5232**	**0.6114**	**0.6587**	**0.3163**	**0.2357**

optimizing for precision@1 with a constrain on total number of parameters.

4.1 Results and Discussion

Table 1 compares the proposed suite of models with the most representative baselines and both SOTA XMLC and TRS models. Inspired by the XMLC evaluation protocols, we focus on short top-K lists (i.e. $K \in \{1, 3, 5\}$). Beyond top-k accuracy, we should note that presented models have different numbers of trainable parameters: PfastreXML, FastXML, and APLC use approximately 350MM due to large transformer networks, Gile uses 12MM, TagSpace 10MM and all four proposed models including LALE use 14MM parameters.

We see that models using label IDs and their names for embedding (proposed hybrid model and the combined baseline approach) outperform other presented baseline methods, including those relying on transformer networks with significantly larger model capacity. For instance, the nDCG@1 of both LALE (0.5232) and Combined (0.5201) is higher than the same metric value of FastXML (0.5147).

This result highlights the potential of using the language structure in label names to achieve higher predictive performance at lower computational cost, which is an important factor in language models with ever increasing capacity [2, 20]. What is more, we have found that the LALE architecture outperforms the model using the full label-level signal (combined baseline) as well as other benchmarks while relying on fewer parameters. We hypothesize that enforcing the joint input-label ID-label description embedding leads to a representation that is easy to process by the classification head, therefore, improving accuracy.

We can observe that even the ID-only baseline can slightly outperform significantly larger networks built on the transformer architecture on nDCG@1. We believe this is for two reasons: (1) the inputs in our dataset are search queries that are significantly shorter than documents used to pre-train the transformer architecture, and (2) the nomenclature used for transformer pre-training differs from the one in search queries. The last finding shows the benefit of training smaller models optimized for a specific problem over massive pre-trained networks.

Table 2: Beyond-accuracy Metrics.

Our Tested Models	**Coverage**	$F1_{Head}$	$F1_{Tail}$
ID-only Baseline	0.208	0.302	0.096
Text-only Baseline	0.281	0.259	0.127
Combined Baseline	0.223	0.318	0.099
LALE	0.302	0.338	0.125

Table 2 presents the labels coverage as well as the binary F1 score on the head and tail filters (see Section 4 for details). The results indicate that the inclusion of textual descriptions of search filters allows the models to better generalize over under-represented tail labels and recommend them more often (high coverage) and more accurately (high tail F1). For example, the text-only and LALE model have higher F1 on three similar tail filters corresponding to power tools (power tools features, wattage, and cord type) indicating information about those labels is shared. This finding is especially prominent when looking at the text-only baseline, where memorizing the most popular label IDs was more difficult. In that case, both coverage and the performance on the tail labels are significantly higher than for counterparts that have access to label IDs. Interestingly, the performance of the combined baseline on tail labels is very similar to the ID-only baseline. We hypothesize that the binary loss function puts higher emphasis on learning about common labels as they appear in more examples and label identifiers provide a better signal about the query-head label mappings and overwhelm the learning process. Our results demonstrate that the proposed approach (LALE) improves model performance for infrequent labels without sacrificing the head accuracy indicating that the new architecture does not suffer from this limitation and making use of label description is promoted.

5 CONCLUSIONS

In this work, we investigate models to find the most relevant search filters to a customer query. To maximize the accuracy of predictions, we propose a new model that leverages search filters' side information (label names) to help learn from and about the less common tail labels. We show that this approach enables computationally efficient, small neural networks to achieve better performance than high capacity transformer networks. Presented models have approximately a factor of twenty fewer parameters than the last generation transformers (BERT), enabling deployment to low-memory architectures (ML on edge) and potentially leading to lowering energy consumption [3].

A potential criticism of the method relying on trainable label information embedding is the high computational cost as each minibatch requires an embedding for each label. This issue, however, can be mitigated by relying on simple models for label embedding (as done here) or subsampling negative labels for each example. Examples of the former are a cornerstone of most massive multi-labels linear classifiers [4, 17] and this approach could be adapted to help accelerate deep learning models.

REFERENCES

[1] Rohit Babbar and Bernhard Schölkopf. 2017. DiSMEC - Distributed sparse machines for extreme multi-label classification. In *WSDM 2017 - Proceedings of the 10th ACM International Conference on Web Search and Data Mining*. Association for Computing Machinery, Inc, 721–729. https://doi.org/10.1145/3018661.3018741 arXiv:1609.02521

[2] Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. On the Dangers of Stochastic Parrots: Can Language Models Be Too Big?. In *FAccT '21: 2021 ACM Conference on Fairness, Accountability, and Transparency, Virtual Event / Toronto, Canada, March 3-10, 2021*, Madeleine Clare Elish, William Isaac, and Richard S. Zemel (Eds.). ACM, 610–623. https://doi.org/10.1145/3442188.3445922

[3] Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. On the Dangers of Stochastic Parrots: Can Language Models Be Too Big?. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency* (Virtual Event, Canada) *(FAccT '21)*. Association for Computing Machinery, New York, NY, USA, 610–623. https://doi.org/10.1145/3442188.3445922

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

[5] Daniel Cer, Yinfei Yang, Sheng yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Céspedes, Steve Yuan, Chris Tar, Yun Hsuan Sung, Brian Strope, and Ray Kurzweil. 2018. Universal sentence encoder for English. In *EMNLP 2018 - Conference on Empirical Methods in Natural Language Processing: System Demonstrations, Proceedings*. 169–174. https://doi.org/10.18653/v1/d18-2029

[6] Patricio Cerda, Gaël Varoquaux, and Balázs Kégl. 2018. Similarity encoding for learning with dirty categorical variables. https://doi.org/10.1007/s10994-018-5724-2ï

[7] Wei Cheng Chang, Hsiang Fu Yu, Kai Zhong, Yiming Yang, and Inderjit S. Dhillon. 2020. Taming Pretrained Transformers for Extreme Multi-label Text Classification. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Association for Computing Machinery, 3163–3171. https://doi.org/10.1145/3394486.3403368 arXiv:1905.02331

[8] Himanshu Jain, Yashoteja Prabhu, and Manik Varma. 2016. Extreme multi-label loss functions for recommendation, tagging, ranking & other missing label applications. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Vol. 13-17-Augu. Association for Computing Machinery, New York, NY, USA, 935–944. https://doi.org/10.1145/2939672.2939756

[9] Sujay Khandagale, Han Xiao, and Rohit Babbar. 2020. Bonsai: diverse and shallow trees for extreme multi-label classification. *Machine Learning* 109, 11 (apr 2020), 2099–2119. https://doi.org/10.1007/s10994-020-05888-2 arXiv:1904.08249

[10] Diederik P. Kingma and Jimmy Ba. 2017. Adam: A Method for Stochastic Optimization. arXiv:1412.6980 [cs.LG]

[11] Yuncheng Li, Yale Song, and Jiebo Luo. 2017. Improving pairwise ranking for multi-label image classification. In *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, Vol. 2017-Janua. Institute of Electrical and Electronics Engineers Inc., 1837–1845. https://doi.org/10.1109/CVPR.2017.199 arXiv:1704.03135

[12] Jingzhou Liu, Wei Cheng Chang, Yuexin Wu, and Yiming Yang. 2017. Deep learning for extreme multi-label text classification. In *SIGIR 2017 - Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 115–124. https://doi.org/10.1145/3077136.3080834

[13] Eneldo Loza Mencía and Johannes Fürnkranz. 2008. Efficient pairwise multilabel classification for large-scale problems in the legal domain. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Vol. 5212 LNAI. 50–65. https://doi.org/10.1007/978-3-540-87481-2_4

[14] Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton Van Den Hengel. 2015. Image-based recommendations on styles and substitutes. In *SIGIR 2015 - Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 43–52. https://doi.org/10.1145/2766462.2767755 arXiv:1506.04757

[15] Bhaskar Mitra and Nick Craswell. 2017. Neural models for information retrieval. arXiv:1705.01509 http://arxiv.org/abs/1705.01509

[16] Nikolaos Pappas and James Henderson. 2018. GILE: A Generalized Input-Label embedding for text classification. https://doi.org/10.1162/tacl_a_00259 arXiv:1806.06219

[17] Yashoteja Prabhu, Anil Kag, Shrutendra Harsola, Rahul Agrawal, and Manik Varma. 2018. Parabel: Partitioned label trees for extreme classification with application to dynamic search advertising. In *The Web Conference 2018 - Proceedings of the World Wide Web Conference, WWW 2018*. Association for Computing Machinery, Inc, 993–1002. https://doi.org/10.1145/3178876.3185998

[18] Yashoteja Prabhu and Manik Varma. 2014. FastXML: A fast, accurate and stable tree-classifier for extreme multi-label learning. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 263–272. https://doi.org/10.1145/2623330.2623651

[19] Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *54th Annual Meeting of the Association for Computational Linguistics, ACL 2016 - Long Papers*, Vol. 3. Association for Computational Linguistics (ACL), 1715–1725. https://doi.org/10.18653/v1/p16-1162 arXiv:1508.07909

[20] Emma Strubell, Ananya Ganesh, and Andrew McCallum. 2019. Energy and Policy Considerations for Deep Learning in NLP. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, Anna Korhonen, David R. Traum, and Lluís Màrquez (Eds.). Association for Computational Linguistics, 3645–3650. https://doi.org/10.18653/v1/p19-1355

[21] Yukihiro Tagami. 2017. AnnexML: Approximate nearest neighbor search for extreme multi-label classification. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Vol. Part F1296. Association for Computing Machinery, New York, NY, USA, 455–464. https://doi.org/10.1145/3097983.3097987

[22] Jason Weston, Samy Bengio, and Nicolas Usunier. 2011. WSABIE: Scaling up to large vocabulary image annotation. In *IJCAI International Joint Conference on Artificial Intelligence*. 2764–2770. https://doi.org/10.5591/978-1-57735-516-8/IJCAI11-460

[23] Jason Weston, Sumit Chopra, and Keith Adams. 2014. #TAGSPACE: Semantic embeddings from hashtags. In *EMNLP 2014 - 2014 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*. Association for Computational Linguistics, Stroudsburg, PA, USA, 1822–1827. https://doi.org/10.3115/v1/d14-1194

[24] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. XLNet: Generalized autoregressive pretraining for language understanding. arXiv:1906.08237 http://arxiv.org/abs/1906.08237

[25] Hui Ye, Zhiyu Chen, Da Han Wang, and Brian D. Davison. 2020. Pretrained Generalized Autoregressive Model with Adaptive Probabilistic Label Clusters for Extreme Multi-label Text Classification. arXiv:2007.02439 http://arxiv.org/abs/2007.02439

[26] Ronghui You, Zihan Zhang, Ziye Wang, Suyang Dai, Hiroshi Mamitsuka, and Shanfeng Zhu. 2018. Attention xml: Label tree-based attention-aware deep model for high-performance extreme multi-label text classification. arXiv:1811.01727 http://arxiv.org/abs/1811.01727

[27] Hsiang-Fu Yu, Kai Zhong, and Inderjit S. Dhillon. 2020. PECOS: Prediction for Enormous and Correlated Output Spaces. arXiv:2010.05878 [cs.LG]

[28] Qi Zhang, Jiawen Wang, Haoran Huang, Xuanjing Huang, and Yeyun Gong. 2017. Hashtag recommendation for multimodal microblog using co-attention network. In *IJCAI International Joint Conference on Artificial Intelligence (IJCAI'17)*. AAAI Press, 3420–3426. https://doi.org/10.24963/ijcai.2017/478

[29] Shuai Zhang, Lina Yao, Aixin Sun, and Yi Tay. 2019. Deep learning based recommender system: A survey and new perspectives. https://doi.org/10.1145/3285029 arXiv:1707.07435

[30] Arkaitz Zubiaga. 2012. Enhancing Navigation on Wikipedia with Social Tags. (2012). arXiv:1202.5469 http://www.flickr.comhttp://arxiv.org/abs/1202.5469