Multilingual Taxonomic Web Page Categorization Through Ensemble Knowledge Distillation

Eric Ye, Xiao Bai, Neil O'Hare, Eliyar Asgarieh, Kapil Thadani, Francisco Perez-Sorrosal, Sujyothi Adiga

***Abstract*—Web page categorization has been extensively studied in the literature and has been successfully used to improve information retrieval, recommendation, personalization and ad targeting. With the new industry trend of not tracking users' online behavior without their explicit permission, using contextual targeting to accurately understand web pages in order to display ads that are topically relevant to the pages becomes more important. This is challenging, however, because an ad request only contains the URL of a web page. As a result, there is very limited available text for making accurate classifications. In this paper, we propose a unified multilingual model that can seamlessly classify web pages in 5 high-impact languages using either their full content or just their URLs with limited text. We adopt multiple data sampling techniques to increase coverage for rare categories in our training corpus, and modify the loss using class-based re-weighting to smooth the influence of frequent versus rare categories. We also propose using an ensemble of teacher models for knowledge distillation and explore different ways to create a teacher ensemble. Offline evaluation shows at least 2.6% improvement in mean average precision across 5 languages compared to a URL classification model trained with single-teacher knowledge distillation. The unified model for both full-content and URL-only input further improves the mean average precision of the dedicated URL classification model by 0.6%. We launched the proposed models, which achieve at least 37% better mean average precision than the legacy tree-based models, for contextual targeting in the Yahoo Demand Side Platform, leading to a significant ad delivery and revenue increase.**

***Index Terms*—contextual targeting, text classification, URL classification, knowledge distillation**

I. Introduction

In recent years, the Internet has been going through a major privacy enhancement. In addition to the restrictions imposed by the EU General Data Protection Regulation (GDPR) and the California Consumer Privacy Act (CCPA), which require explicit user permission for web services to track users' online behavior, many browsers including Safari, Firefox and Chrome are also starting to restrict the use of advertising cookies on websites. This trend poses a challenge to the traditionally successful approach of *behavioral* ad targeting, because collecting users' historical online activities with browser cookies and using them to derive users' interests and recommend relevant ads would only be possible for a small fraction of users. Therefore, *contextual* targeting, which serves ads only based on the context information available at the time of an ad call, is becoming increasingly critical to online advertising.

Among the different emerging contextual targeting solutions [1], *category-based contextual targeting* [2] has proved to be an impactful alternative in the new cookieless world. In this approach, advertisers specify a number of topics from a pre-defined taxonomy and their ads are eligible to be displayed on web pages relevant to these topics. Accurate prediction of the topic categories of web pages is key to the success of category-based contextual targeting. However, although web page categorization has been extensively studied in the literature [3]–[5], applications to contextual targeting encounter three unique challenges that have not been properly addressed.

First, the number of categories is relatively large. The popular IAB Content Taxonomy[1] developed for contextual targeting and brand safety has approximately 700 categories over 4 tiers. In this work, we use the Yahoo Interest Category (YIC), which was developed specifically for contextual targeting and consists of 442 categories over 5 tiers. In contrast, most existing work [6] [7] classifies web pages to a handful of top-level topic categories. Moreover, when considering a large number of categories in a rich taxonomy, a web page may be characterized by multiple categories—including multiple *top-level* categories—for different advertising purposes. For instance, a web page about news for a luxury retailer[2] can be categorized as both *Content & Entertainment/News/Business & Finance* and *Retail/Product/Clothing/Luxury Clothing*. Thus, web page categorization for contextual targeting is a multi-label classification problem over a large number of categories.

Second, web pages are highly skewed in terms of categories, with a small number of frequent categories and a long tail. The hierarchical nature of the taxonomy further increases the skew as categories at higher tiers are more frequent than their descendants. This makes it difficult to create a training dataset that represents the entire taxonomy without a large class imbalance. In addition, given the importance of the international market for online advertising, non-English web pages must also be classified. However, the skewed proportions of English and non-English web pages in advertising data creates an additional imbalance for training that needs to be accounted for by the web page classifier.

Third, while existing work typically uses the full content of web pages for categorization, such information is not always available for contextual targeting. An ad request coming to

Eric Ye, Xiao Bai, Neil O'Hare, Eliyar Asgarieh, and Francisco Perez-Sorrosal are affiliated with Yahoo Research, Mountain View, California, USA. Kapil Thadani is affiliated with Yahoo Research, New York, USA. Sujyothi Adiga is affiliated with Yahoo Inc., Bangalore, India. E-mail: {jiayunye, xbai, nohare, eliyar.asgarieh, thadani, fperez, sujyothi}@yahooinc.com

[1] https://iabtechlab.com/standards/content-taxonomy/

[2] https://finance.yahoo.com/news/post-bankruptcy-ahead-neiman-marcus-050125178.html

an ad system usually contains the URL of the web page on which an ad will be displayed, along with other context information such as device, location and time. A modern ad system receives billions of ad requests each day but can only selectively crawl a portion of the web pages due to the high cost. An effective contextual targeting solution requires a classifier that can not only categorize crawled web pages using their full content, but also accurately classify web pages using only the text found in the URLs.

To address these challenges, we propose the first (to the best of our knowledge) unified Transformer-based model that accurately categorizes multilingual web pages with either full content or only URLs into multiple labels in a large taxonomy. We achieve this by first fine-tuning large multilingual Transformer models with human-annotated web pages using their full content, and then distilling the knowledge from an ensemble of these large models into a small Transformer model using an augmented dataset designed for both web pages with full content and those with only URLs. This approach bridges the performance gap caused by the input (full content vs URL) that was also observed by an early attempt which uses expanded tokens in the URLs for an SVM-based multi-label web page classifier [8]. The inherent label skew over a large taxonomy is addressed by our loss re-weighting scheme that simultaneously smooths data imbalance between positive and negative samples as well as between frequent and rare categories, in addition to a number of dedicated data augmentation techniques.

Note that the challenges listed above were also partially addressed by the models proposed in the conference version of this work [2]. We provide additional improvements through ensemble knowledge distillation and model unification with respect to different inputs. More specifically, the main contributions of this work are as follows:

- We adapt Transformer models to multi-label classification by modifying the output classification layer and propose novel class-based loss re-weighting and data sampling techniques to deal with label skew, achieving 37% higher mean average precision than legacy classifiers.
- We achieve multilingual classification for content in 5 target languages by augmenting human labeled training datasets with machine translated samples, leading to at least 0.9% higher mean average precision over models trained purely with human-labeled datasets.
- We train a single unified model that predicts multiple class labels for web pages with full content and those with only URLs by altering the training data to adapt to both types of input and distilling knowledge from models trained with full content. Offline evaluation shows 0.6% better mean average precision than a dedicated URL-only model distilled from the same teacher. The unified model also achieves 12.7% higher mean average precision than a URL model directly fine-tuned using human-labeled datasets without knowledge distillation.
- We design a customizable multi-teacher knowledge distillation training framework and explore two types of teacher ensembles: *Ensemble-Aggregate* aggregates the predictions from multiple teacher models as the soft label of a sample, while *Ensemble-Best* takes the teacher model with the best accuracy on a language to generate the soft labels for the samples in that language. We experiment various ways of building individual teacher models by varying random initialization, loss-specific hyperparameters, and language-specific training data when fine-tuning pre-trained language models for multi-label web page categorization. Comparison with the URL model trained with single-teacher knowledge distillation shows 2.6% higher mean average precision.
- We deploy the proposed model to support category-based contextual targeting at Yahoo, and show through a series of launches and online metrics how each key model improvement positively influences ad delivery and revenue.

II. Related Work

In this section, we summarize relevant prior literature in a number of areas that relate to this work. As this work is an extension of our prior study [2], we also highlight the main difference between the new models proposed here and their alternatives from the previous work.

Multi-label Text Classification. Hierarchical multi-label classification (HMC) has been extensively studied in machine learning. In HMC, categories or labels are organized in a hierarchical structure [9], while multi-label classification allows assigning multiple labels to each document [10], [11]. In HMC, one primary approach is to train independent binary classifiers for each category. The other, which we focus on, is to train a single multi-label model capable of predicting all categories simultaneously.

Multilingual Text Classification. Early research in multilingual document classification focused on cross-lingual sentence representations using parallel corpora [12]–[14]. However, recent advances in multilingual masked language models, like Multilingual BERT [15], XLM [16], and XLM-RoBERTa [17], have improved multilingual text classification. Our work differs by focusing on multi-label taxonomic classification with numerous classes while enabling classification based solely on providing the model with a URL—which may contain short or non-meaningful text snippets—thereby setting it apart from approaches that use more extensive textual content.

URL Classification. Classifying URLs without accessing the web page content is of increasing importance. Applications that require classification before crawling (e.g., focused crawling [18]) or cannot afford the high latency or high cost of large-scale crawling (e.g., contextual targeting based on topics of URL [2]) require lightweight and accurate models to classify web pages only using the information available in their URLs. Early attempts on URL classification [8], [19], [20] suffer from a significant performance gap compared to models that have access to the full web page content, as they rely solely on features like tokens and n-grams [8], [19], [20] drawn from the URLs. To better understand the semantics of URLs, we distill knowledge from an ensemble of Transformer models fine-tuned with full web page content, which significantly improves the model directly fine-tuned with URLs. Singh et al. [21] also build an ensemble for URL classification but the ensemble is applied at inference time. This is much more costly than our multi-teacher ensemble as we only use a single (student) model for inference. Moreover, our model can classify both web pages with full content and with URLs alone, which none of the existing models are designed for.

Weight Redistribution. Weight redistribution has been explored in machine learning to correct class imbalance, biased datasets or corrupted labels [22]–[24]. Many modern machine learning frameworks such as Scikit, PyTorch and TensorFlow offer utilities to pre-compute class weights and allow a weight re-scaling factor to be incorporated in loss functions[3]. Prior research has extended these ideas to *online* class re-weighting, e.g., by minimizing the loss on a clean unbiased validation set using a meta-gradient descent step on the weights of the current mini-batch [25]. Class weights are typically defined as proportional to inverse class frequency, but recent work has shown that smoothing these weights produces better empirical outcomes [26], [27]. Smoothing can be accomplished via heuristic transformations of inverse class frequency weights (e.g., square root [26], log [28]), as well as formulations that can interpolate between uniform and inverse-frequency weights [27]; the latter is better suited to taxonomic categorization as hyperparameters can be tuned to trade off performance on head versus tail categories. In this work, we propose a more intuitive weight redistribution strategy to correct extreme class imbalances in taxonomic multi-label classification using two hyperparameters that balance the loss among classes while simultaneously balancing the loss between positive and negative examples.

Pre-trained Language Models. In recent years, models based on the Transformer architecture [29], which uses a *self-attention* mechanism, have driven significant advances on a variety of tasks such as language generation, translation, question-answering and classification. Some examples of recent models that build on this architecture include BERT [30], RoBERTa [31], GPT [32] [33] and DistilBERT [34]. These models are pre-trained on a large unsupervised document corpus and subsequently fine-tuned on a supervised downstream task [30]. We follow this approach and fine tune pre-trained language models for hierarchical multi-label classification.

Multi-Teacher Knowledge Distillation. Knowledge distillation is a model compression technique for transferring knowledge from larger deep neural networks into smaller ones. Multi-teacher distillation has shown that different teacher architectures can provide their own useful knowledge, which can be transferred to the student model to improve knowledge distillation. In a typical teacher-student framework [35], the teacher is often a large model or an ensemble of large models, and a common approach is to utilize the averaged response from all teacher networks as the supervision signal. This technique has proven to be effective in training the student model as it incorporates knowledge in the form of response-based representations [36]–[39], feature-based representations [40]–[42] or a combination of both [43]. In this work, we explore different variations of teacher ensembles for response-based knowledge distillation. Our results demonstrate a significant improvement in model performance when compared to the models produced by single-teacher knowledge distillation.

Category-Based Contextual Targeting. Contextual targeting [1] is an advertising strategy that displays ads relevant to the content of a web page. Category-based [44] contextual targeting is widely used in digital advertising to match ad content with relevant website content based on specific categories. This strategy aims to ensure that ads are displayed in contexts that align with the advertiser's target audience and goals [44], [45]. A typical approach involves classifying web content into predefined categories, such as sports, entertainment, technology, or finance, and then delivering ads that are most relevant to each category. Hashemi [4] summarizes the major web page classification approaches using text, images or both. To the best of our knowledge, this is the first study to propose a web page classification model that solely relies on the URL itself for contextual targeting.

This paper can be viewed as an enhancement of the preliminary models presented in the conference version of this work [2] with the following major extensions: (i) We improve the proposed web page categorization model by introducing a customizable multi-teacher ensemble training framework; (ii) We develop a unified modeling approach that allows web page categorization with or without full content, in English and non-English languages, with a single model; (iii) We launch the new, unified model in a production environment and show its significant impact on ad delivery and revenue in category-based contextual targeting for Yahoo's demand-side advertising platform, driven by the algorithmic improvements made in this extension.

III. Unified Web Page Categorization

To support category-based contextual targeting at scale, it is important to classify both crawled web pages with full content and un-crawled web pages with only URLs. We introduce in this section our unified multilingual web page categorization model that can seamlessly classify both kinds of web pages from 5 targeted languages. We formulate the task as a multi-label classification problem and describe the adaptation of pre-trained Transformer encoders from the BERT [30] family to address it. We provide details on the development of our multilingual training and evaluation corpus. We then present our multi-teacher knowledge distillation framework that makes models compact while outperforming the standard knowledge distillation using a single teacher model. Finally, we describe how we adapt the training process to allow a single distilled model to classify both crawled and uncrawled web pages.

A. Problem Formulation

Given a taxonomy of M categories $\{c_1, c_2, ...c_M\}$ and the text available for a web page x_i, the objective is to predict a probability vector $\hat{y}_i = [\hat{y}_{i,c_1}, \hat{y}_{i,c_2}, ..., \hat{y}_{i,c_M}]$, where $\hat{y}_{i,c_j}$ is the probability of category c_j being relevant to web page x_i. In this work we use the Yahoo Interest Categories (YIC) taxonomy, which is a hierarchical taxonomy containing 442

[3] e.g.https://pytorch.org/docs/master/generated/torch.nn.BCEWithLogitsLoss.html

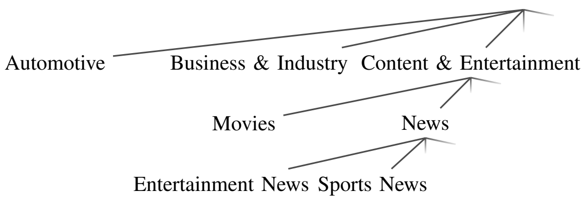


Fig. 1: Example categories from the YIC taxonomy.

interest categories: 12 tier-1 (root-level) categories, 100 tier-2 categories, 259 tier-3 categories, 66 tier-4 categories and 5 tier-5 categories. Figure 1 shows a few examples of YIC categories. The hierarchical structure implies that a web page assigned to any category (e.g., "Content & Entertainment/News") would also be categorized to its ancestor categories (e.g., "Content & Entertainment"). A page may also be described by multiple categories, e.g., a car blog by "Automotive" and "Content & Entertainment/News".

The following sections describe how we adapt a popular transfer learning framework to this multi-label classification problem. Our basic approach is to fine tune a pre-trained Transformer model as the shared encoder and and use an output layer with sigmoid activations for each category to predict whether a web page belongs to that category (Section III-B). We rely on professional editors to annotate web pages with relevant taxonomic categories and devise various techniques to build annotated datasets (Section III-C). As shown by our experiments in Section IV, a naive attempt at using editorial data to train such models is sub-optimal and the accuracy of Transformer models typically improves significantly as the model size (i.e., inner dimensionality, number of layers, number of self-attention heads) is increased, but larger models are impractical for large-scale classification tasks due to significantly higher computational cost. We address this limitation through knowledge distillation [36], a framework in which the predictions of an accurate but expensive model are used to train a lightweight distilled model. In this work, we propose to use a *teacher ensemble* to distill knowledge from multiple teacher models to a lightweight student model (Section III-D). We also present an approach to augment the datasets to allow a single distilled student model to classify both crawled and un-crawled web pages (Section III-E).

B. Modeling Approach

We fine-tune a pre-trained Transformer encoder to predict the probability vector $\hat{y}_i$. The encoder layers are shared among all categories. For each category c_j, an output layer with sigmoid activation is trained to minimize the corresponding binary cross-entropy loss. The loss for a web page x_i is computed as a weighted sum of its losses on all M categories. Using $y_{i,c_j} \in \{0,1\}$ to indicate whether page x_i belongs to category c_j, given the training set of N samples, the total loss can be computed as follows:

$$L = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{M} w_{c_j} y_{i,c_j} \log \hat{y}_{i,c_j} + (1-y_{i,c_j})\log(1-\hat{y}_{i,c_j}) \tag{1}$$

where the weights for negative labels (i.e., $y_{i,c_j} = 0$) are implicitly always 1, and the weights for positive labels (i.e., $y_{i,c_j} = 1$) are defined for each category c_j as follows:

$$w_{c_j} = \mu \frac{\max_k f_{c_k} + \gamma N}{f_{c_j} + \gamma N} \tag{2}$$

where $f_{c_j} = \sum_{i=1}^{N} \hat{y}_{i,c_j}$ denotes the frequency of category c_j in the training data, $\mu \in \mathbb{R}$ controls the category-agnostic weight given to positive samples and $\gamma \in \mathbb{R}_{\geq 0}$ is a smoothing factor scaled by the number of training samples N. This proposed weighting scheme simultaneously addresses data imbalance between positive & negative samples via μ, and between frequent & rare categories via γ. As $\gamma \to 0$, the weights become inversely proportional to class frequency (i.e., $w_{c_j} \propto 1/f_{c_j}$), while as $\gamma \to \infty$, all positive samples receive a uniform weight $w_{c_j} = \mu$. This ability to interpolate between uniform and inverse frequency weighting is similar to the class-balanced loss formulation of [27], but features a more intuitive smoothing factor γ and greater control due to μ.

C. Annotated Corpus Development

For our initial work on corpus development, we collected a traffic-based stratified sample of English language bid request URLs from the Yahoo Demand-side Platform (DSP) during a 6-month period from Jan 2020 to July 2020. All data labelling was carried out internally by an in-house editorial team. Given a web page in this dataset, we crawl the HTML, extract the page title and body, strip HTML tags and present it to editors to receive one or more category labels from the taxonomy[4].

The long tail category distribution, however, means that a random sample of bid request pages has very low coverage of torso and tail categories. For example, 27 categories have no labelled pages in a 15k random sample, while 114 categories have less than 5 labelled pages in the same sample. For this reason, we adopt two targeted sampling approaches to address this problem: (i) URL collection, and (ii) active learning.

- *URL Collection*. Since many of the target categories have zero or very few labelled samples in our initial random sample, we did not have data to bootstrap a model that could be used to assist with data collection. For this reason, in collaboration with the Yahoo editorial team, we used a method that we refer to as *URL Collection* or *UC*, where the editorial team is given a set of categories and asked to find URLs from diverse websites that are relevant to those categories. After these candidate URLs have been collected, they are then fully annotated with respect to additional taxonomy categories that they are relevant to. Although this is clearly a biased form of data collection, since data does not come from the population of bid request URLs, our results will demonstrate that this is nevertheless a very useful way to bootstrap models for rare categories.
- *Active learning*. Active learning is a method for using model predictions to sample documents for annotation.

[4] Using the crawled content instead of accessing a page directly by its URL avoids any inconsistency caused by page content update.

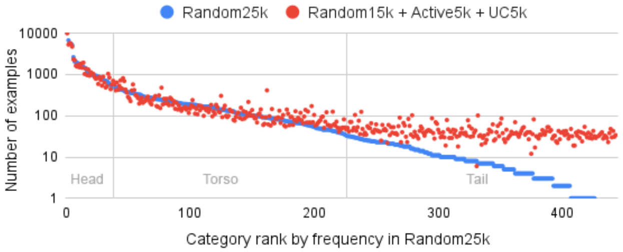


Fig. 2: Pages per category in a random sample and in a combined sample with URL Collection and active learning.

After first bootstrapping data with URL Collection, we can train initial models for tail and torso categories. To gather additional candidate pages for these rare categories, we adopt the simple approach of sampling pages for which the model score is higher than a threshold, and these pages are then manually labelled by our editorial team. During corpus construction, this process of random sampling followed by targeted sampling was iterated a number of times, each time based on a recent 6-month stratified sample. Figure 2 shows how this improves coverage for tail categories.

In addition to English, we identified 4 other target languages for web page classification: Spanish, French, Portuguese and Traditional Chinese. We created language specific corpora for each language by sampling web pages with a mix of stratified random sampling, URL Collection and active learning for each of the targeted languages. These web pages were annotated with the same taxonomy labels by professional editors who are fluent speakers of these languages. In addition, we use the Google Translate API[5] to translate the content of the annotated English web pages to each of the non-English languages and use them as part of the annotated datasets for training the models. The evaluation datasets for the non-English languages do not contain any translated data.

The corpus was partitioned into training, development and test sets for each target language as follows:

- *Training Set*, containing a mix of data sampled randomly, by URL Collection, and by active learning, as described above.
- *Development Set*, used to make initial decisions on optimal hyperparameters, and for model selection via early stopping. This data is a random subset of the stratified random sample of DSP bid request URLs.
- *Test Set*, held-out dataset, which serves as a gatekeeper that determines whether the model can be deployed in production. If the model passes the overall quality requirements agreed upon, it is put into production. Like the development set, this is a random subset of the stratified random sample.

Corpus statistics for each target language can be found in Table I. *Trainable categories* is defined as the number of categories with at least one positive example in the train set. *Testable categories* are defined as the number of categories with at least one positive example in the development set and test set respectively. As the development and test sets only contain stratified random samples, some long tail categories may not be covered due to their limited sizes. This is also our motivation for using URL collection and active learning to ensure the models are trained to predict all categories in the taxonomy with enough examples (Figure 2).

TABLE I: Statistics of the annotated corpus. #S - Number of samples. #C - Number of trainable or testable categories.[6]

Language	Train		Dev		Test	
	#S	#C	#S	#C	#S	#C
English	56k	442	5k	362	13k	391
Spanish	48k	442	1.2k	280	7k	378
French	48k	442	1.2k	299	8k	384
Portuguese	48k	442	1.2k	286	8k	380
Traditional Chinese	48k	442	1.2k	257	8k	387

D. *Multi-Teacher Knowledge Distillation*

Directly fine tuning pre-trained Transformer models using human annotated datasets usually requires large model sizes to achieve good classification accuracy. Larger models however require higher computational resources and longer inference time, making them impractical to classify pages at web scale. To address this issue, we propose a multi-teacher knowledge distillation approach. Distillation can be accomplished using a variety of techniques. In our work, knowledge is transferred to the student model through a large dataset of unlabeled examples. Instead of using a single model as the teacher, we use an ensemble of multiple models to transfer their knowledge to the student model.

Specifically, in our multi-teacher knowledge distillation framework, we first train K (K$\geq$1) teacher models using a human labeled dataset. We refer to these models as the *teacher ensemble*. Note that each teacher model is a multilingual model that can classify documents in any of our target languages. We then use the probability vectors from these models' output to produce a soft label (i.e., a probability vector $y_i^s = [y_{i,c_1}^s, y_{i,c_2}^s, ..., y_{i,c_M}^s]$) for each sample x_i in a transfer set. Finally, using the transfer set, a student model is trained to optimize the cross entropy loss described in Formula (1) using the soft labels y_{i,c_j}^s as ground truth instead of the binary labels y_{i,c_j}.

1) Teacher Model Training.

All teacher models are trained using the human labeled dataset described in Table I. In addition to the page title and body extracted from the HTML, we also parse the URL (e.g., `https://news.yahoo.com/sports/football`) and extract the domain (e.g., `news.yahoo.com`) and path (e.g., `/sports/football`) from the URL to use in the input. The URL domain, path and page title are designated as the first segment while the page body constitutes the second segment when fine tuning the pre-trained Transformer based teacher models. Using full content (i.e., URL, title and body) leads to higher accuracy than using only URL text as we will see in Table VII (T-Rand-1 vs. XML-R-Large$_u$). Since we target a multilingual scenario, we use XLM-RoBERTa-Large [17] as

[5] https://translate.google.com

[6] Categories with at least 1 positive sample in the dataset.

our backbone model in this work. We train K models with different settings to create the teacher ensemble.

We consider three different ways of training a set of K teacher models to create a teacher ensemble:

- *Random initialization variants*. Different random seeds are used to initialize the parameters in the output layer, a common approach to create neural network ensembles [36].
- *Loss re-weighting variants*. Different hyperparameter combinations μ and α (Equation (2)) are used to train teacher models that achieve different balance between positive-negative samples and frequent-rare categories.
- *Language variants*. Different subsets of the training dataset (i.e., each subset contains samples specific to one selected language) are used to fine tune multilingual teacher models that perform better in one specific language.

2) Transfer Set Development

In response-based knowledge distillation, the knowledge of the teacher model is transferred to a student model by training the student model to match the teacher model's soft targets (class probabilities) using a transfer set. Our transfer set consists of two sets of web pages: (1) a large random sample of *unlabeled* web pages collected with traffic-based stratified sampling on the bid request URLs from the Yahoo DSP in a 6-month period from July 2021 to December 2021; (2) the same web pages in our human annotated training set in section III-C. Instead of using the human provided labels, we generate soft labels for these web pages using the teacher ensemble approaches we present in the next section and use them as part of the transfer set for knowledge distillation.

3) Teacher Ensemble Approaches.

We consider two different ways of generating the soft labels $y_i^s = [y_{i,c_1}^s, y_{i,c_2}^s, ..., y_{i,c_M}^s]$ for each sample x_i in the transfer set, using the multiple teacher models trained with one of the approaches described in Section III-D1.

- *Ensemble-Aggregate*. Following the literature on response-based knowledge distillation, given a sample x_i, we apply each of the K teacher models to predict a probability vector $y_i^{(k)} = [y_{i,c_1}^{(k)}, y_{i,c_2}^{(k)}, ..., y_{i,c_M}^{(k)}]$, $k = 1, 2, ..., K$. We use category-wise average or maximum to represent the probability for a category. Specifically, the soft label for sample x_i and category c_j can be computed as

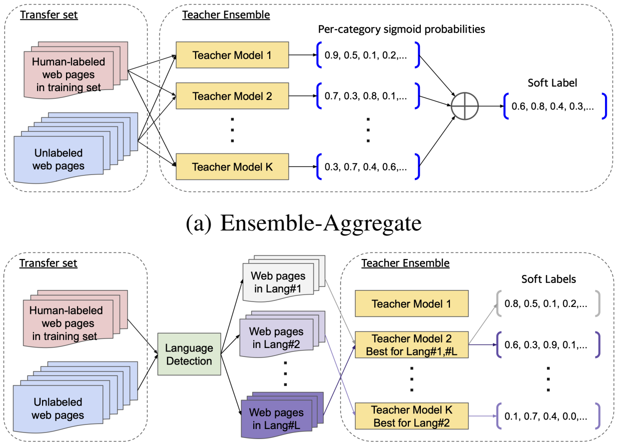
$$y_{i,c_j}^s = \frac{1}{K} \sum_k y_{i,c_j}^{(k)} \tag{3}$$

or

$$y_{i,c_j}^s = \max_k y_{i,c_j}^{(k)} \tag{4}$$

Figure 3(a) illustrates the Ensemble-Aggregate approach by averaging the category-wise predictions from the teacher models.

- *Ensemble-Best*. Ensemble-Aggregate requires predicting categories for each sample K times to generate a soft label. Depending on the size of the transfer set, this process may be cost inefficient. An alternative is to use a single model out of the K teacher models that leads to the best performance on a specific language to generate soft labels for samples in that language. Language of a web page can be detected using signals from its URL and the ad request. We do not elaborate on language detection in this work. Practically, we perform an evaluation of K teacher models using a development dataset to determine the best teacher model for each language. The soft label for a web page in a language is simply the probability vector predicted by the best model for that language. Note that the same teacher model may be the best performing model for one or more languages. Figure 3(b) illustrates the Ensemble-Best approach.



(a) Ensemble-Aggregate

(b) Ensemble-Best

Fig. 3: Teacher ensemble approaches.

Most previous work on knowledge distillation has been in the multi-class setting, where temperature scaling is applied to smooth the predicted label distribution from the teacher to ensure that predictions for classes outside of the sole positive class have non-trivial values. In the multi-label classification setting, where predictions are not normalized over all classes, the benefit of this smoothing is not clear. Our preliminary experiments showed that temperature of 1 (i.e., no scaling) performed best. We thus use the teacher predictions directly without any scaling when generating the soft labels.

4) Student Model Training.

Once the soft labels are generated for the transfer set, we train the student model by fine tuning a pre-trained Transformer model that is smaller than the teacher models using the transfer set as discussed in section III-D2 When optimizing the cross entropy loss described in Formula (1), the soft labels y_{i,c_j}^s are used as ground truth instead of the binary labels y_{i,c_j} for both datasets. We use this process to distill large models like XLM-RoBERTa-Large (561M parameters) into models like XLM-RoBERTa-Base (279M parameters), resulting in more computationally efficient inference at scale.

E. Data Augmentation for a Unified Student Model

Our teacher models are trained with the data annotated by professional editors (Section III-C). When generating the soft labels for the transfer set, we follow the same process to crawl the web pages and construct the input using URL domain and path, page title and body extracted from the HTML for the teacher models.

When training the student model, we aim at building a unified model that can categorize both crawled and un-crawled web pages. This is motivated by two key observations: (1) crawling to extract page content may add significant latency, in addition to consuming a lot of resources, when running contextual targeting at scale; (2) there is often sufficient information in URL domains (e.g., `news.yahoo.com`) and paths (e.g., `/sports/football`) to produce reasonable categories. Therefore, we create two training samples from each labeled web page: the first sample uses the URL, title and body as input and its soft label as target; the second sample uses the URL as input and the same soft label as target. When fine-tuning a pre-trained Transformer student model, we add the prefix token `[Content]` to the input of each sample that has crawled title and body and add the prefix token `[URL]` to the input of each sample that only has URL. During inference, we again append the appropriate prefix token to the input.

IV. Results

In this section, we present the offline evaluation of the proposed web page classification model. We first evaluate the different techniques for addressing data imbalance (Section IV-B) and show how machine translation helps to better classify multilingual web pages (Section IV-C). We then delve into our unified multilingual web page categorization model, emphasizing the performance of the teacher ensembles (Section IV-D) and the student models distilled from them (Section IV-E). We finally report the ablation studies on the impact of teacher ensemble size, the impact of transfer set size, and the effect of unified input (Section IV-F).

A. Experimental Setup

Corpus. We build the train, development and test datasets as described in Section III-C. The train set consists of a stratified sample of web pages included in the ad requests collected by Yahoo DSP (Demand Side Platform) in 5 languages (English, Spanish, French, Portuguese, Traditional Chinese), with additional English web pages collected by human-editors and active learning for rare categories, and English web pages translated into each of the target languages. The development and test sets only consist of a stratified sample of web pages for each language. Corpus statistics for each target language can be found in Table I. Note that the development set is only used to evaluate the teacher models in Table V and Figure 5. All other evaluation relies on the test set.

Metric. We use a macro average mAP (Mean Average Precision) as our main, threshold-independent, metric. Macro mAP is computed as the mean of each category's average precision[7]. All models were trained over the entire set of 442 categories, but due to the skewed category distribution in the test set, evaluation is limited to categories with at least one positive test example, which we refer to as "*testable*" categories.

Implementation. We modify the open source code of Hugging Face `transformers`[8] to support multi-label output and loss re-weighting, as described in Section III-A. Hyperparameters are optimized using grid search and the best models are selected based on mAP using early stopping on the development set. Experiments were conducted using eight Nvidia A100 GPUs with 80GB of VRAM each.

B. Evaluation of Imbalanced Data Classification

A key contribution of this work is to address the inherent class imbalance in web page classification through category-based loss re-weighting and targeted data collections as described in Section III-A and Section III-C. In this section, we evaluate their impact using a single large-size model trained with the full content from English web pages, in order to rule out the impact of multilingual datasets, url-only input, and knowledge distillation.

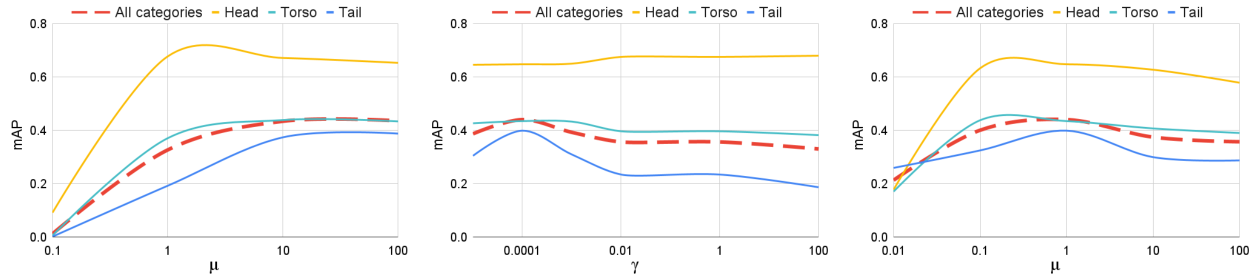
1) Loss Re-Weighting

Since preliminary results showed that RoBERTa-Large [31] models outperformed similarly sized BERT models, consistent with published results, we use RoBERTa-Large as our English language model for the experiments in this section. Table II shows baseline results for RoBERTa-Large models evaluated on the entire set of 391 *testable categories* on the English language test set. These models are compared against a unigram+bigram XGBoost [46] model trained on the same data. We use XGBoost as a baseline here as this was used as our legacy production model, and XGBoost has been shown to perform well on a variety of tasks compared with other decision tree models [46]. Optimal values for the μ and γ weighting hyperparameters are selected based on mAP on the development set. The results for models trained for 5 epochs show the benefit of loss re-weighting: a vanilla implementation without re-weighting achieves an mAP of 0.326, which is worse than the baseline XGBoost model, and far behind that of the best mAP of 0.440 when the proposed class-based re-weighting is used. Interestingly, *positive class weighting*, which assigns the same weight (empirically, always greater than 1) to all the positive samples and a default value of 1 to all the negative samples, achieves results almost on par with the more sophisticated *class-based weighting*. Although the early work on BERT models reported task-specific fine tuning results for a small number of epochs [30], our preliminary results suggested that training for longer can lead to significant improvements. The results in Table II show that the mAP for the best model improves significantly, from 0.440 to 0.462, with longer training. Longer training also appears to make these models more robust to the choice of loss, as the difference between the unweighted vs weighted models is much smaller in this case. Overall, apart from the 5-epoch RoBERTa model with no re-weighting, all models show dramatic improvements over the baseline XGBoost model.

Figure 4 examines the impact of the re-weighting hyperparameters μ and γ in more details for models trained for

[7] We do not report micro average mAP because its value is dominated by frequent categories that are easier to predict. Our focus is also to improve relatively rare categories as they may be important for specific advertising needs.

[8] https://huggingface.co/docs/transformers/index



(a) Varying μ w/o per-class re-weighting ($\gamma = \infty$) (b) Varying γ w/o positive weight ($\mu = 1$) (c) Varying μ at optimal $\gamma = 0.0001$

Fig. 4: Analysis of the effect of positive class weight μ and smoothing factor γ with RoBERTa-Large.

TABLE II: mAP for various weighting strategies for English web page classification with RoBERTa-Large on the test set

Model	5 epochs	80 epochs
XGBoost	0.337	
Transformer models		
No re-weighting	0.326	0.450
Positive-class weighting	0.435	0.460
Class-based weighting	0.440	0.462

5 epochs. In addition to mAP averaged over all categories, for this analysis we split the taxonomy into head, torso and tail categories based on the number of training samples per category in a 25k stratified random sample, as follows: (i) *head*: categories with $>=$500 examples, (ii) *torso*: categories with $>=$30 and $<$500 examples, (iii) *tail*: categories with $<$30 examples. Figure 4a examines the effect of the positive weight factor μ without per-class re-weighting: while a default value of 1 is optimal for the head categories, this leads to poor performance on tail categories and sub-optimal performance on torso categories, with an optimal value for torso/tail at around $\mu = 10$. This is because torso and tail categories suffer much more from the positive-negative imbalance problem compared to head categories. Increasing the value of μ encourages the model to focus on improving torso and tail categories. Figure 4b examines the effect of the smoothing factor when the positive labels have no additional weight ($\mu = 1$). For larger values of the γ hyperparameter (i.e. *less* re-weighting), we see that torso/tail categories perform poorly, with torso/tail performance improving as re-weighting is applied by reducing γ. According to Formula (2), as γ gets smaller, the smoothing term γN gets larger, and thus the positive samples from the torso and tail categories receive higher weights. Figure 4c shows the effect of changing μ when an approximately optimal smoothing factor is used: while the trend is the same, the interaction of the two hyperparameters leads to a lower optimal value for μ.

2) Targeted Data Collection

In this section, we delve into the implications of the targeted data sampling techniques outlined in Section III-C. In these experiments, we train the RoBERTa-Large models using different subsets of the training set, specifically comparing datasets comprising 15k, 20k, and 25k documents, which are assembled through a mix of random sampling, URL Collection, and active learning data acquisition strategies. The results, as presented in Table III, reveal a modest enhancement across all segments when introducing additional random data to the initial 15k dataset. In contrast, the inclusion of URL Collection data leads to a substantial improvement in the case of tail categories and a modest improvement for torso categories. As URL Collection primarily focuses on rare categories, it underscores the effectiveness of this strategy, even though it may exhibit bias. Moreover, after training a model using URL Collection data, we accumulate sufficient data to bootstrap models for torso and tail categories, enabling the application of active learning techniques to sample these specific categories. The final three rows of Table III illustrate the impact of further expanding the size of the training corpus, demonstrating that active learning sampling provides additional enhancements over URL Collection, particularly for tail categories.

TABLE III: mAP for RoBERTa-Large on English web page classification using various sampling methods on the test set. Random - Random sample. UC - URL Collection. Active - Active learning.

Sampling Strategy	#samples	All	Head	Torso	Tail
Random15k	15k	0.390	0.652	0.439	0.269
Random20k	20k	0.397	0.659	0.450	0.271
Random15k + UC5k	20k	0.447	0.652	0.452	0.394
Random25k	25k	0.401	0.655	0.451	0.282
Random20k + UC5k	25k	0.445	0.647	0.448	0.393
Random15k + Active5k + UC5k	25k	0.452	0.649	0.448	0.413

C. Evaluation of Multilingual Models

Our ultimate goal is to train a single multilingual model that is able to classify web pages in 5 target languages. To this end, we first evaluate in this section a number of large-size models trained on our multilingual dataset described in Section III-C. Models trained with the best dataset variant serve as the base models of the proposed teacher ensembles that we will evaluate in Section IV-D. To have a single model that is able to predict over multiple languages, we employ XLM-RoBERTa-Large, a state-of-the-art model known for its effectiveness in multilingual NLP tasks [17]. We compare the performance of XLM-RoBERTa-Large models trained with human-annotated editorial data, translated data, and a combination of both in Table IV. Notably, training a non-English classifier exclusively using translated data yields competitive results compared to training with data directly annotated in the target language. These findings are promising because using automatically

TABLE IV: mAP for multilingual models with various training datasets on the test set. E - Editorial data. T - Translated data.

Model	Train Language	en	es	fr	pt	zh-tw
Monolingual						
RoBERTa-Large	E:en	0.460	-	-	-	-
XLM-R-Large	E:en	0.458	-	-	-	-
Multilingual						
XLM-R-Large	E:en + T:es/fr/pt/zh-tw	0.447	0.544	0.519	0.517	0.485
XLM-R-Large	E:en/es/fr/pt/zh-tw	0.468	0.555	0.552	0.536	0.532
XLM-R-Large	E+T:es/fr/pt/zh-tw	-	0.550	0.529	0.536	0.520
XLM-R-Large	E+T:en/es/fr/pt/zh-tw	0.474	0.577	0.557	0.560	0.543

translated data not only augments our training dataset without requiring additional human effort but also allows us to leverage English-language URL Collection and active learning strategies to enhance coverage of rare categories. The most optimal results, encompassing all languages, including English, are achieved by combining translated and directly annotated data, leading to improvements in mAP of 1.3% for English, 4.0% for Spanish, 0.9% for French, 4.5% for Portuguese, and 2.1% for Traditional Chinese. Surprisingly, these results reveal that when evaluating on English data, the multilingual model trained with data from five languages not only matches the performance of English-only models but surpasses them with a relative improvement of 3.0% over the English-only RoBERTa-Large model and 3.5% over an English-only XLM-RoBERTa-Large model. Similarly, Table IV indicates that adding English data while fine-tuning the XLM-RoBERTa-Large multilingual model significantly improves mAP for all the non-English languages.

D. Evaluation of Teacher Ensembles

In Section II we presented three variants of creating individual teacher models (i.e., *Random initialization*, *Loss reweighting* and *Language*) and two types of teacher ensemble approaches (i.e., *Ensemble-Aggregate* and *Ensemble-Best*). We evaluate their combinations in this section. Table V summarizes the performance of individual teachers as well as the corresponding teacher ensembles on the development dataset. All the teacher models are fine-tuned from the same XML-RoBERTa-Large model.

The top section in Table V (Random initialization variants) reports the performance of 5 individual teacher models trained with different random initialization, denoted as **T-Rand-k** ($1 \leq k \leq 5$), two variants of Ensemble-Aggregate that aggregate the per category predictions from the 5 teacher models using average probability (**T-Rand-Avg**) or max probability (**T-Rand-Max**), and one variant of Ensemble-Best that uses the best model among the 5 teachers for each language (**T-Rand-Best**). We observe that the 3 teacher ensembles clearly outperform all the 5 individual models on the 5 targeted languages. While the improvement is expected for T-Rand-Best as it explicitly picks the best performing model, T-Rand-Avg and T-Rand-Max achieve better mAP than T-Rand-Best. This implies different teacher models may capture complementary knowledge. By aggregating their predictions, there is also a regularization effect that reduces the biases from individual teacher models. T-Rand-Avg achieves the best mAP, which improves the performance of the baseline single teacher model (T-Rand-1 as presented in the conference version of this work [2]) by 7.1% for English, 4.8% for Spanish, 3.2% for French, 7.2% for Portuguese and 4.7% for Traditional Chinese.

The middle section in Table V (Loss re-weighting variants) reports the performance of 5 individual teacher models trained with different loss re-weighting hyperparameters, denoted as **T-Hyper-k** ($1 \leq k \leq 5$), two variants of Ensemble-Aggregate that aggregate the per category predictions from the 5 teacher models using average probability (**T-Hyper-Avg**) or max probability (**T-Hyper-Max**), and one variant of Ensemble-Best that uses the best model among the 5 teacher models for each language (**T-Hyper-Best**). Note that T-Hyper-1 and T-Rand-1 are the same individual teacher model which is also the same as the teacher model in [2]. We name it differently in Table V for the easy of comparison within the same group of models. We observe that all 3 ensemble models improve the mAP of each individual teacher models across the five targeted languages. The performance of T-Hyper-Avg is 2.1% better than that of T-Rand-Avg.

The bottom section in Table V (Language variants) explores the language variants of the teacher models. Here, we train "monolingual" teacher models using samples in one specific language to fine tune an XLM-RoBERTa-Large model for that language. We obtain 5 individual teacher models, one for each target language. As expected, the mAP on each target language outperforms the mAPs on the four off-the-target languages[9]. The teacher ensembles are created in the same way as **T-Lang-Avg** and **T-Lang-Max** for Ensemble-Aggregate and as **T-Lang-Best** for Ensemble-Best. All 3 teacher ensembles lead to higher mAP than each individual model. More importantly, the best performing language specific teacher ensemble T-Lang-Avg also achieves higher mAP than the baseline individual teacher model T-Rand-1 for four out of five languages.

As shown by Table V, all the three variants of building individual teacher models allow creating teacher ensembles that can categorize web pages more accurately than an individual model. Teacher ensembles based on loss re-weighting perform slightly better than those based on random initialization, and they outperform the teacher ensembles based on language. Ensemble-Aggregate by averaging per category probability to create soft labels (i.e., T-*-Avg) is the best among all ensemble approaches (compared to T-*-Max and T-*-Best).

E. Evaluation of Student Models

Table VI summarizes the performance of XLM-RoBERTa-Base student models trained using the Ensemble-Aggregate and Ensemble-Best teacher ensembles, evaluated on the test dataset. We use the transfer dataset labeled by each teacher ensemble to train two separate models: *Content Model* is trained using the full content as input and *URL Model* is trained using the URL as input. Our transfer dataset consists of the human labelled dataset plus 300K web pages randomly sampled from Yahoo DSP for each language.

We observe that the student models distilled from teacher ensembles have higher mAP than students models distilled

[9] Although the fine-tuning is done with samples in one language, since the pre-trained model is multilingual, the performance on the off-the-target languages remains competitive.

TABLE V: mAP of teacher models on the development set. Highlighted entries indicate the best mAP per language overall (**$\underline{\text{bold underlined}}$**), per section (**bold**), and among individual teacher models per section ($\underline{\text{underlined}}$).

Teacher	Model Type	μ	γ	en	es	fr	pt	zh-tw	Avg
Random initialization variants									
T-Rand-1 [2]	Individual teacher	0.1	0.01	0.466	0.478	$\underline{0.494}$	0.456	0.387	0.456
T-Rand-2	Individual teacher	0.1	0.01	0.477	$\underline{0.481}$	0.488	0.457	0.380	0.456
T-Rand-3	Individual teacher	0.1	0.01	$\underline{0.483}$	0.477	0.480	0.464	0.382	0.457
T-Rand-4	Individual teacher	0.1	0.01	0.473	0.479	0.487	$\underline{0.467}$	0.382	0.457
T-Rand-5	Individual teacher	0.1	0.01	0.477	0.467	0.486	0.459	$\underline{0.388}$	0.455
T-Rand-Best	Ensemble-Best	-	-	0.483	0.481	0.494	0.467	0.388	0.462
T-Rand-Max	Ensemble-Aggregate	-	-	0.491	0.489	0.496	0.471	0.394	0.468
T-Rand-Avg	Ensemble-Aggregate	-	-	**0.499**	**0.501**	**0.510**	**$\underline{0.489}$**	**0.405**	**0.472**
Loss re-weighting variants									
T-Hyper-1 [2]	Individual teacher	0.1	0.01	0.466	$\underline{0.478}$	$\underline{0.494}$	0.456	0.387	0.456
T-Hyper-2	Individual teacher	0.01	0.01	0.463	0.460	0.477	0.454	0.373	0.445
T-Hyper-3	Individual teacher	1.0	0.01	0.474	0.473	0.484	0.466	0.383	0.456
T-Hyper-4	Individual teacher	2.0	0.01	0.471	0.467	0.480	0.457	0.388	0.453
T-Hyper-5	Individual teacher	0.1	0.001	$\underline{0.481}$	0.476	0.486	$\underline{0.471}$	$\underline{0.388}$	0.442
T-Hyper-Best	Ensemble-Best	-	-	0.481	0.478	0.494	0.471	0.388	0.462
T-Hyper-Max	Ensemble-Aggregate	-	-	0.482	0.491	0.508	0.476	0.397	0.471
T-Hyper-Avg	Ensemble-Aggregate	-	-	**$\underline{0.500}$**	**$\underline{0.502}$**	**$\underline{0.515}$**	**0.488**	**$\underline{0.407}$**	**$\underline{0.482}$**
Language variants									
T-Lang-En	Individual teacher	2.0	0.01	$\underline{0.459}$	0.427	0.448	0.415	0.358	0.422
T-Lang-Es	Individual teacher	2.0	0.01	0.443	$\underline{0.452}$	0.443	0.416	0.341	0.419
T-Lang-Fr	Individual teacher	2.0	0.01	0.428	0.423	$\underline{0.449}$	0.405	0.321	0.405
T-Lang-Pt	Individual teacher	2.0	0.01	0.439	0.428	0.425	$\underline{0.424}$	0.344	0.412
T-Lang-Zh-Tw	Individual teacher	2.0	0.01	0.420	0.420	0.417	0.383	$\underline{0.367}$	0.402
T-Lang-Best	Ensemble-Best	-	-	0.459	0.452	0.449	0.424	0.367	0.430
T-Lang-Max	Ensemble-Aggregate	-	-	0.472	0.470	0.474	0.459	0.379	0.451
T-Lang-Avg	Ensemble-Aggregate	-	-	**0.484**	**0.484**	**0.492**	**0.468**	**0.391**	**0.464**

TABLE VI: mAP for the student models on the test set. c - Model trained with full content (i.e., URL+Title+Body) as input. u - Model trained with URL as input.

Model	Teacher	en	es	fr	pt	zh-tw	Avg
Teacher/Teacher Ensemble Models							
T-Hyper-1 [2]	-	0.474	0.577	0.557	0.560	0.543	0.542
T-Hyper-Avg	-	0.504	0.607	0.588	0.590	0.580	0.574
Student Content Models - trained and tested with full content as input							
S-Rand-1$_c$ [2]	T-Rand1	0.483	0.581	0.564	0.570	0.548	0.549
S-Rand-Best$_c$	T-Rand-Best	0.489	0.589	0.564	0.566	0.560	0.554
S-Rand-Avg$_c$	T-Rand-Avg	**0.500**	**0.599**	**0.586**	**0.581**	**0.573**	**0.569**
S-Hyper-Best$_c$	T-Hyper-Best	0.485	0.591	0.564	0.551	0.534	0.545
S-Hyper-Avg$_c$	T-Hyper-Avg	0.493	0.591	0.576	0.572	0.566	0.559
Student URL Models - trained and tested with URL as input							
S-Rand-1$_u$ [2]	T-Rand1	0.423	0.508	0.497	0.506	0.418	0.470
S-Rand-Best$_u$	T-Rand-Best	0.424	0.523	0.498	0.508	0.416	0.474
S-Rand-Avg$_u$	T-Rand-Avg	**0.427**	**0.521**	**0.508**	**0.513**	0.424	**0.479**
S-Hyper-Best$_u$	T-Hyper-Best	0.424	0.510	0.488	0.496	0.410	0.466
S-Hyper-Avg$_u$	T-Hyper-Avg	0.423	0.518	0.494	0.508	**0.426**	0.474

from an individual teacher. For the student content models, the mAPs are even higher than those of the individual teacher T-Hyper-1 [2]. The mAP scores are also very close to the best teacher ensemble T-Hyper-Avg (Table V). Student content models S-Rand-Avg$_c$ and S-Hyper-Avg$_c$ outperform S-Rand-Best$_c$ and S-Hyper-Best$_c$. This implies better teacher ensembles lead to better student models. The student models **S-Rand-Avg**$_c$ and **S-Rand-Avg**$_u$, distilled from the teacher ensemble T-Rand-Avg, have the highest mAP among all student models, improving mAP the baseline model S-Rand-1$_c$ [2] and S-Rand-1$_u$ [2] by 3.6% and 1.9% on average across 5 languages.

Table VII compares the best student URL model distilled from the teacher ensemble S-Rand-Avg$_u$ with models directly fine-tuned from pre-trained models using URL as input. We observe that URL-only inference using a model trained with content yields poor mAP and a model fine-tuned specifically with URL-only input shows significant improvement. Nevertheless, using the proposed multi-teacher knowledge distillation to train a base-sized URL model can achieve significantly higher mAP (12.7% on average) than a large model directly fine-tuned with URL data across all the languages.

TABLE VII: mAP for different URL models on the test set. XML-R-Base$_u$ and XLM-R-Large$_u$ are fine-tuned URL models without using knowledge distillation.

Model	Train	Test	en	es	fr	pt	zh-tw	Avg
T-Rand-1 [2]	Content	URL	0.309	0.380	0.381	0.368	0.306	0.349
XLM-R-Base$_u$	URL	URL	0.345	0.433	0.415	0.430	0.351	0.397
XLM-R-Large$_u$	URL	URL	0.374	0.470	0.449	0.455	0.376	0.425
S-Rand-Avg$_u$	URL	URL	**0.427**	**0.521**	**0.508**	**0.513**	**0.424**	**0.479**

A main contribution of this work is that our student model is a unified multilingual model that can categorize both full content and URL-text only as input. Table VIII compares the unified student model (**S-Rand-Avg**$_m$) with the separate student models. S-Rand-Avg$_m$ achieves competitive mAPs as the student content and URL models distilled from the same teacher ensemble T-Rand-Avg. On average, the mAP is 0.9% lower than S-Rand-Avg$_c$ and 0.6% higher than S-Rand-Avg$_u$. This model is better than the student models distilled from a single teacher model, i.e., S-Rand-1$_c$ and S-Rand-1$_u$ for web pages with and without full content, by 2.7% and 2.6% respectively. More importantly, with the unified model, we only need to deploy one single model which halves the memory footprint in our system. Note that although we augment the transfer set and double its size for the unified model, it is still trained for the same number of steps as the separate models. This keeps the training cost comparable.

TABLE VIII: mAP for the unified student model on the test set. m - Unified model trained with mixed inputs.

Model	Train Input	en	es	fr	pt	zh-tw	Avg
Unified Student Models - tested with full content as input							
S-Rand-1_c [2]	Content	0.483	0.581	0.564	0.570	0.548	0.549
S-Rand-Avg_c	Content	0.500	**0.599**	**0.586**	**0.581**	**0.573**	**0.569**
S-Rand-Avg_m	Content & URL	**0.502**	0.596	0.575	0.580	0.567	0.564
Unified Student Models - tested with URL as input							
S-Rand-1_u [2]	URL	0.423	0.508	0.497	0.506	0.418	0.470
S-Rand-Avg_u	URL	0.427	**0.521**	**0.508**	0.513	0.424	0.479
S-Rand-Avg_m	Content & URL	**0.433**	0.519	0.506	**0.522**	**0.429**	**0.482**

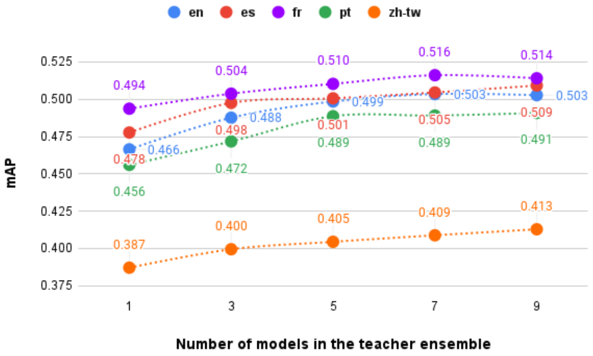


Fig. 5: Impact of teacher ensemble size. T-Rand-Avg is evaluated on the development set.

F. Ablation Study

1) Impact of Teacher Ensemble Size

In this section, we evaluate how the number of teacher models in an ensemble influences its performance as well as that of the student model distilled from it. Figure 5 shows the mAP per language for teacher ensembles with 1, 3, 5, 7 and 9 models respectively. We focus on T-Rand-Avg because the corresponding student models have the best performance (Table VI). We observe that as the number of models in the teacher ensemble increases, the mAP also increases.

Table IX compares the performance of the unified student models distilled from teacher ensemble with sizes 5 and 9. We observe that using 9 models in T-Rand-Avg leads to 1.4% and 1.2% higher mAPs on average for the student model. This implies different teacher model can have complementary knowledge and different correctness on the same sample and combine all together could lead less bias and provide better supervision for the student model learning. Despite the superior accuracy, using more models in the teacher ensemble implies higher computational cost for predicting probability vectors for each sample in the transfer set, especially when the transfer set is large.

2) Impact of Transfer Set Size

Table VI and Table VII show that a base-size distilled model trained with a transfer set achieves significantly higher mAP than a large-size model directly fine-tuned with a human labeled set for both web pages with and without full content. Table X reports how the size of the transfer set influences the mAP of the unified student models (S-Rand-Avg_m). As described in Section III-D2, our transfer set is composed of web pages from the human labeled training set (Section III-C) and those randomly sampled from an unlabeled set representing the live advertising traffic. The student models in Table X are trained for 1M steps using soft labels generated by the teacher ensemble T-Rand-Avg for web pages from editorial data, randomly sampled data, or a combination of both.

TABLE IX: Impact of teacher ensemble size on student models. S-Rand-Avg_m is evaluated on the test set.

Test Input	#Models	en	es	fr	pt	zh-tw	Avg
Content	5	0.502	0.596	0.575	0.580	0.567	0.564
Content	9	**0.503**	**0.613**	**0.584**	**0.591**	**0.571**	**0.572**
URL	5	0.433	0.519	0.506	0.522	0.429	0.482
URL	9	**0.434**	**0.533**	**0.514**	**0.526**	**0.433**	**0.488**

TABLE X: Impact of transfer set size on the unified student models S-Rand-Avg_m, evaluated on the test set. Soft labels for web pages in the editorial labeled data (E) and additional random unlabeled data (R). Numbers in the #R column are per language: e.g. 50k × 5 indicates 50k each for 5 languages.

#E	#R	en	es	fr	pt	zh-tw	Avg
Unified Student Model - tested with full content as input							
248k	0	0.471	0.583	0.554	0.564	0.549	0.544
248k	50k × 5	0.487	0.590	0.570	0.574	0.558	0.556
248k	100k × 5	0.496	0.598	0.574	0.580	0.559	0.561
248k	300k × 5	0.502	0.596	0.575	0.580	0.567	0.564
0	300k × 5	**0.504**	**0.597**	**0.579**	**0.581**	**0.567**	**0.566**
Unified Student Model - tested with URL as input							
248k	0	0.401	0.489	0.478	0.472	0.402	0.448
248k	50k × 5	0.417	0.510	0.495	0.500	0.416	0.468
248k	100k × 5	0.427	0.512	0.497	0.509	0.422	0.473
248k	300k × 5	**0.433**	**0.519**	**0.506**	**0.522**	**0.429**	**0.482**
0	300k × 5	0.434	0.505	0.492	0.501	0.405	0.467

We observe from Table X that training directly with soft labels from the editorial data leads to an increase in mAP compared to not using knowledge distillation (Table V and Table VII). Introducing random data labelled by the teacher model for distillation leads to further performance gains. The improvement is more significant with a larger amount of random samples, especially for web pages without full content. Interesting, using enough random samples only achieves the highest mAP for web pages with full content. This may be because the label distribution is better represented by this large random set. Yet, since using the full transfer set (of web pages from the editorial set and the random set) leads to much better mAP for web pages with only URL, our final unified model is trained with the full transfer set.

3) Impact of Prefix Token in Student Models

When training the unified student model for classifying web pages with full content or with only URLs, we enhance the training data by adding a prefix token, `[Content]` or `[URL]`, to the input of each sample as described in Section III-E. We assess the performance of the unified student models trained with or without the prefix tokens. The results in Table XI demonstrate that the unified student model (**S-Rand-Avg**$_m$), trained on input data with a prefix token for each sample, achieves higher mAPs regardless of whether a web page has full content or not during the inference.

V. Product Impact on Contextual Targeting

Our taxonomic web page classification models were developed to support category-based contextual targeting in the Yahoo DSP (Demand Side Platform). We built a Spark Streaming pipeline on AWS that uses our unified multilingual

TABLE XI: Impact of prefix token on the unified student models. S-Rand-Avg$_m$ trained with or without prefix tokens are evaluated on the test set.

Test Input	prefix	en	es	fr	pt	zh-tw	Avg
Content	no	0.497	0.598	0.580	0.582	0.567	0.565
Content	yes	**0.503**	**0.613**	**0.584**	**0.591**	**0.571**	**0.572**
URL	no	0.430	0.523	0.507	0.520	0.431	0.482
URL	yes	**0.434**	**0.533**	**0.514**	**0.526**	**0.433**	**0.488**

TABLE XII: Post launch metrics for contextual targeting. The relative improvement for the contribution percentage of contextual targeting to DSP before and after launch are reported. #Models - number of models required.

#Models	Post launch coverage		Relative change w.r.t. pre-launch		
	Market	Uncrawled	Impression	Click	Revenue
1	en	No	+56%	+17%	+77%
2	en	Yes	+257%	+194%	+353%
4	en/es/fr/pt/zh-tw	Yes	+37%	+31%	+33%
1	en/es/fr/pt/zh-tw	Yes	+19%	+36%	+17%

model to categorize the web pages in the ad requests to the DSP in near real time. Categories assigned to a web page are filtered using per category thresholds that ensure a precision of at least 0.8 on a standalone evaluation dataset randomly sampled from the online traffic. URLs and their predicted categories are stored in a key-value store for real-time lookup during ad serving. When an ad request arrives in the DSP, ads that target at least one of the web page's categories are eligible for the ad auction.

In this section, we report the performance of four representative model launches in production for contextual targeting during the course of building the product. We measure the contribution of contextual targeting to the entire Yahoo DSP before and after a model launch for impressions, clicks and revenue. These contributions are measured 15 days before and 15 days after each launch date, and the relative improvement for each metric is summarized in Table XII. Relative changes are computed to de-emphasize temporal effects.

The first launch replaced the production XGBoost model with our Transformer-based model for crawled English web pages. The XGBoost model consists of one binary classifier for each category, trained using words and Wikipedia entities from the web page content as features. We observe that the Transformer-based model increased the contribution of Contextual Targeting to DSP by 56% for impressions, 17% for clicks, and 53% for revenue.

One major contribution of this work is a distilled Transformer-based model that can accurately classify uncrawled web pages solely based on tokens from the URLs. When we launched this model into production for English pages, in addition to the model that only classifies crawled web pages, the contribution of contextual targeting to DSP impressions increased by 257%. Given that a significant fraction of web pages do not have their content available for analysis, and therefore could not previously be classified, this launch enabled classification of a far greater number of documents, and so greatly increased the impact of contextual targeting.

The third launch involved the two distilled multilingual models that extend our contextual targeting solution to crawled and uncrawled web pages in Spanish, French, Portuguese, and Traditional Chinese. As expected, compared to the earlier models that only classify English pages, this launch increased the contribution of Contextual Targeting to DSP by 37% for impressions, 31% for clicks, and 33% for revenue.

Note that the second and the third launches consist of separate models for classifying web pages with crawled content and web pages with only URLs respectively. The forth launch replaces the models in the third launch with a single unified model that categorizes crawled and uncrawled web pages seamlessly. Specifically, we replaced the 4 running models for crawled English, crawled non-English, uncrawled English and uncrawled non-English web pages with 1 single model. This reduces the memory footprint of the model and the maintenance complexity. More importantly, as the accuracy of the unified model improves, after launching the new model into production, we observe 19% increase in impressions, 36% increase in clicks, and 17% increase in revenue for Yahoo's category-based contextual targeting on DSP.

Together, these post-launch metrics show that each of these launches, based on model variations described in this paper, contributed significantly to the growth of contextual targeting within the DSP platform at Yahoo.

VI. Conclusion

In this paper, we proposed, for the first time, a unified multilingual model that accurately categorizes web pages using either full content or only URLs. We showed through extensive evaluation that (i) URL Collection (i.e. tasking editors with actively searching for web pages relevant to rare categories) is critical to bootstrap models for torso/tail categories, which further enables the use of active learning sampling, to address the skewed category distribution; (ii) augmenting multilingual data through machine translation significantly improves the classification accuracy for both English and non-English pages; (iii) class-based loss re-weighting is important to improve classification accuracy for rare categories; (iv) knowledge distillation allows us to train lightweight and more accurate models, especially when page content is not crawled for URLs; (v) multi-teacher knowledge distillation is a crucial approach for improving the supervision signals from the teachers' responses; (vi) augmenting the training set with different input formats and proper prefix tokens is a simple yet effective approach to enable a unified student model to classify both crawled and uncrawled web pages. This model has been successfully deployed into the production system to improve the category-based contextual targeting in Yahoo's DSP.

As future improvements, we are expanding the model to cover more languages that are important to our international markets. Besides, as the web continuously evolves and the categories need to reflect emerging advertising needs, adapting the model to evolving interest taxonomy with minimum relabeling and re-training remains a challenge.

ACKNOWLEDGMENTS

We would like to thank all the anonymous editors from the Yahoo editorial team, led by Samantha Giordano, Katy Kuo, Anna Howell and Karen Wu, to evaluate the datasets used in this work.

REFERENCES

[1] K. Zhang and Z. Katona, "Contextual advertising," *Marketing Science*, vol. 31, no. 6, pp. 980–994, 2012.

[2] E. Ye, X. Bai, N. O'Hare, E. Asgarieh, K. Thadani, F. Perez-Sorrosal, and S. Adiga, "Multilingual taxonomic web page classification for contextual targeting at yahoo," in *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2022, pp. 4372–4380.

[3] X. Qi and B. D. Davison, "Web page classification: Features and algorithms," *ACM Comput. Surv.*, vol. 41, no. 2, feb 2009.

[4] M. Hashemi, "Web page classification: a survey of perspectives, gaps, and future directions," *Multimedia Tools and Applications*, vol. 79, pp. 11 921–11 945, 2020.

[5] A. Onan, "Classifier and feature set ensembles for web page classification," *Journal of Information Science*, vol. 42, no. 2, pp. 150–165, 2016.

[6] H. Yu, J. Han, and K. C.-C. Chang, "Pebl: Positive example based learning for web page classification using svm," in *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2002, pp. 239–248.

[7] D. Shen, Z. Chen, Q. Yang, H.-J. Zeng, B. Zhang, Y. Lu, and W.-Y. Ma, "Web-page classification through summarization," in *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2004, pp. 242–249.

[8] M.-Y. Kan, "Web page classification without the web page," in *Proceedings of the 13th International World Wide Web Conference on Alternate Track Papers & Posters*, 2004, pp. 262–263.

[9] C. N. Silla and A. A. Freitas, "A survey of hierarchical classification across different application domains," *Data Mining and Knowledge Discovery*, vol. 22, no. 1-2, pp. 31–72, 2011.

[10] G. Tsoumakas and I. Katakis, "Multi-label classification: An overview," *International Journal of Data Warehousing and Mining*, vol. 3, no. 3, pp. 1–13, 2007.

[11] K. Bhatia, H. Jain, P. Kar, M. Varma, and P. Jain, "Sparse local embeddings for extreme multi-label classification," in *Advances in Neural Information Processing Systems*, 2015, pp. 730–738.

[12] M. Artetxe and H. Schwenk, "Massively Multilingual Sentence Embeddings for Zero-Shot Cross-Lingual Transfer and Beyond," *Transactions of the Association for Computational Linguistics*, vol. 7, pp. 597–610, 2019.

[13] H. Schwenk and X. Li, "A corpus for multilingual document classification in eight languages," in *Proceedings of the Eleventh International Conference on Language Resources and Evaluation*, 2018.

[14] S. Ruder, I. Vulić, and A. Søgaard, "A survey of cross-lingual word embedding models," *Journal of Artificial Intelligence Research*, vol. 65, pp. 569–630, 2019.

[15] T. Pires, E. Schlinger, and D. Garrette, "How multilingual is multilingual BERT?" in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019, pp. 4996–5001.

[16] A. Conneau and G. Lample, "Cross-lingual language model pretraining," in *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, vol. 32, 2019, pp. 7059–7069.

[17] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, E. Grave, M. Ott, L. Zettlemoyer, and V. Stoyanov, "Unsupervised cross-lingual representation learning at scale," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 8440–8451.

[18] H. Lu, D. Zhan, L. Zhou, and D. He, "An improved focused crawler: Using web page classification and link priority evaluation," *Mathematical Problems in Engineering*, 2016.

[19] M.-Y. Kan and H. O. N. Thi, "Fast webpage classification using url features," in *Proceedings of the 14th ACM International Conference on Information and Knowledge Management*, 2005, pp. 325–326.

[20] E. Baykan, M. Henzinger, L. Marian, and I. Weber, "A comprehensive study of features and algorithms for url-based topic classification," *ACM Trans. Web*, vol. 5, no. 3, 2011.

[21] N. Singh, N. S. Chaudhari, and N. Singh, "Online url classification for large-scale streaming environments," *IEEE Intelligent Systems*, vol. 32, no. 2, pp. 31–36, 2017.

[22] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: Synthetic minority over-sampling technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, 2002.

[23] D. Hendrycks, M. Mazeika, D. Wilson, and K. Gimpel, "Using trusted data to train deep networks on labels corrupted by severe noise," in *Advances in Neural Information Processing Systems*, vol. 31, 2018, pp. 10 477–10 486.

[24] M.-L. Zhang and K. Zhang, "Multi-label learning by exploiting label dependency," in *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2010, pp. 999–1008.

[25] M. Ren, W. Zeng, B. Yang, and R. Urtasun, "Learning to reweight examples for robust deep learning," in *Proceedings of the 35th International Conference on Machine Learning*, 2018, pp. 4334–4343.

[26] D. Mahajan, R. Girshick, V. Ramanathan, K. He, M. Paluri, Y. Li, A. Bharambe, and L. van der Maaten, "Exploring the limits of weakly supervised pretraining," in *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.

[27] Y. Cui, M. Jia, T. Lin, Y. Song, and S. Belongie, "Class-balanced loss based on effective number of samples," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Los Alamitos, CA, USA: IEEE Computer Society, jun 2019, pp. 9260–9269. [Online]. Available: https://doi.ieeecomputersociety.org/10.1109/CVPR.2019.00949

[28] K. R. M. Fernando and C. P. Tsokos, "Dynamically weighted balanced loss: Class imbalanced learning and confidence calibration of deep neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 7, pp. 2940–2951, 2022.

[29] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, vol. 30, 2017, pp. 6000–6010.

[30] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2019, pp. 4171–4186.

[31] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized bert pretraining approach," arXiv:1907.11692., 2019.

[32] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners," 2019.

[33] T. B. Brown, B. P. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krüger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. J. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, "Language models are few-shot learners," in *Advances in Neural Information Processing Systems*, 2020, pp. 1877–1901.

[34] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter," arXiv:1910.01108, 2019.

[35] J. Gou, B. Yu, S. J. Maybank, and D. Tao, "Knowledge distillation: A survey," *International Journal of Computer Vision*, vol. 129, no. 6, pp. 1789–1819, 2021.

[36] G. Hinton, O. Vinyals, J. Dean *et al.*, "Distilling the knowledge in a neural network," arXiv:1503.02531, 2015.

[37] B. B. Sau and V. N. Balasubramanian, "Deep model compression: Distilling knowledge from noisy teachers," arXiv:1610.09650, 2016.

[38] Y.-Y. Yang, Y.-A. Lin, H.-M. Chu, and H.-T. Lin, "Deep learning with a rethinking structure for multi-label classification," arXiv:1802.01697, 2018.

[39] Y. Gao, T. Parcollet, and N. D. Lane, "Distilling knowledge from ensembles of acoustic models for joint ctc-attention end-to-end speech recognition," in *2021 IEEE Automatic Speech Recognition and Understanding Workshop*, 2021, pp. 138–145.

[40] A. Romero, N. Ballas, S. E. Kahou, A. Chassang, C. Gatta, and Y. Bengio, "Fitnets: Hints for thin deep nets," arXiv:1412.6550, 2014.

[41] S. Park and N. Kwak, "Feed: Feature-level ensemble for knowledge distillation," arXiv:1909.10754, 2019.

[42] A. Wu, W.-S. Zheng, X. Guo, and J.-H. Lai, "Distilled person re-identification: Towards a more scalable system," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 1187–1196.

[43] S. You, C. Xu, C. Xu, and D. Tao, "Learning from multiple teacher networks," in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2017, pp. 1285–1294.

[44] A. Broder, M. Fontoura, V. Josifovski, and L. Riedel, "A semantic approach to contextual advertising," in *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, 2007, pp. 559–566.

[45] A. Bahirat, "Contextual recommendations using nlp in digital marketing," in *Proceedings of Sixth International Congress on Information and Communication Technology*, 2022, pp. 655–664.

[46] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, pp. 785–794.

Eric Ye is a principal research engineer at Yahoo Research, Mountain View, USA. He received his Master in Library and Information Science from Fu Jen Catholic University, New Taipei City, Taiwan. His research interests include machine learning, natural language processing, query understanding and information retrieval.

Xiao Bai is a principal research scientist at Yahoo Research, Mountain View, USA. She received her PhD in Computer Science from INRIA, Rennes, France. Her research interests include machine learning, natural language processing, information retrieval, and distributed algorithms. Her contributions to various domains of research have been presented in top venues where she regularly serves as a PC or SPC member, such as SIGIR, CIKM, WSDM and the Web conference.

Neil O'Hare is a principal research scientist at Yahoo Research, San Francisco, USA. He received his PhD in Computer Science in Dublin City University, Ireland in 2007. His research interests include machine learning, natural language processing and information retrieval. Neil has published over 40 publication at international conferences and journals, and regularly serves as a PC member at conferences such as SIGIR, WSDM and the Web Conference.

Eliyar Asgarieh is a principal research engineer at Yahoo Research, Mountain View, USA. He received his PhD from Tufts University, Boston, MA, focused on adaptive identification and modeling of nonlinear time-varying systems. His work is concentrated on machine learning research and implementation, including natural language processing, information retrieval, text classification, and ranking.

Kapil Thadani is a senior research scientist at Yahoo Research in New York, NY, USA. He received his PhD in Computer Science from Columbia University in the City of New York in 2015. His research interests include natural language processing and deep learning. Kapil has published over 25 peer-reviewed articles in international conferences and journals, and regularly serves as a PC member for conferences such as ACL, NAACL and EMNLP.

Francisco Perez-Sorrosal is a principal research engineer at Yahoo Research, San Francisco, USA. He received his PhD in Computer Science from Universidad Politecnica de Madrid in 2009. His research interests include, among others, Distributed/Complex Systems, NLP, and AI. He has published peer-reviewed papers in well-known international conferences and journals, such as The Web Conference, ACL, and VLDB Journal.

Sujyothi Adiga is a principal software engineer at Yahoo, Bangalore, India. She received her Master in Computer Application from Mangalore University, India. Her research interests include building cost effective large distributed systems for natural language processing, contextual targeting and ad ranking.