NLP Service APIs and Models for Efficient Registration of New Clients (Appendix)

A Reproducibility/Implementation Details

In this section we provide details about the dataset, architecture and training procedures used for each of the three tasks. We provide the datasets used, code, hyperparameters for all the tasks in the code submitted along with the submission.

A.1 NER

We use the standard splits provided in the Ontonotes dataset (Pradhan et al., 2007). Our codebase builds on the official PyTorch implementation released by (Devlin et al., 2018). We finetune a cased BERT base model with a maximum sequence length of 128 tokens for 3 epochs which takes 3 hours on a Titan X GPU.

A.2 Sentiment Classification

As described previously, we use the Amazon dataset (Ni et al., 2019). For each review, we use the standard protocol to convert the rating to a binary class label by marking reviews with 4 or 5 stars as positive, reviews with 1 or 2 stars as negative and leaving out reviews with 3 stars. We randomly sample data points from each domain to select 1000, 200 and 500 positive and negative reviews each for the train, validation and test splits, respectively. We leave out the domains that have insufficient examples, leaving us with 22 domains. We use the finetuning protocol provided by the authors of (Sun et al., 2019) and use the uncased BERT base model with a maximum sequence length of 256 for this task. We train for 5 epochs (which takes 4 hours on a Titan X GPU) and use the validation set accuracy after every epoch to select the best model.

A.3 Auto Complete Task

We use 20NewsGraoup dataset while regarding each content class label as a client. We remove header, footer from the content of the documents and truncate the size of each client to around 1MB. We use word based tokenizer with a vocabulary restricted to top 10,000 tokens and demarcate sentence after 50 tokens. The reported numbers in Table 3 are when using TF-IDF vector for domain sketch. We diIn this section, we reportd not evaluate other kinds of domain sketch on this task. We train all the methods for 40 epochs with per epoch train time of 4 minutes on a Titan X GPU.

We adopt the tuned hyperparameters corresponding to PTB dataset to configure the baseline Melis et al. (2020). Since the salience information from the client sketch can be trivially exploited in perplexity reduction and thereby impede learning desired hypothesis beyond trivially copying the salience information, we project the sketch vector to a very small dimension of 32 before fanning it out to the size of vocabulary. We did not use any non-linearity in G_ϕ and also employ dropout on the sketches.

B Details of MoE method (Guo et al., 2018)

MoE employs a shared encoder and a client specific classifier. We implemented their proposal to work with our latest encoder networks. Our implementation of their method is to the best of our efforts faithful to their scheme. The only digression we made is in the design of discriminator: we use a learnable discriminator module that the encoder fools while they adopt MMD based metric to quantify and minimize divergence between clients. This should, in our opinion, only work towards their advantage since MMD is not sample efficient especially given the small size of our clients.

OOD Clients	OOD		ID	
	Base	KYC	Base	KYC
BC/CCTV + BC/Phoenix	63.8	70.1	86.00	86.7
BN/PRI + BN/VOA	88.7	91.6	84.5	86.2
NW/WSJ + NW/Xinhua	73.9	79.2	80.8	82.2
BC/CNN + TC/CH	78.3	80.4	85.6	87.1
WB/Eng + WB/a2e	76.2	78.9	86.4	87.5
Average	76.2	80.0	84.7	85.9

Table 7: Performance on the NER task on the Ontonotes dataset when using TF-IDF as the client sketch.

C Results with Different Client Sketches

In this section we provide results on every OOD split for the different client sketches described in Section 3 along with more details.

- **TF-IDF**: This is a standard vectorizer used in Information Retrieval community for document similarity. We regard all the data of the client as a

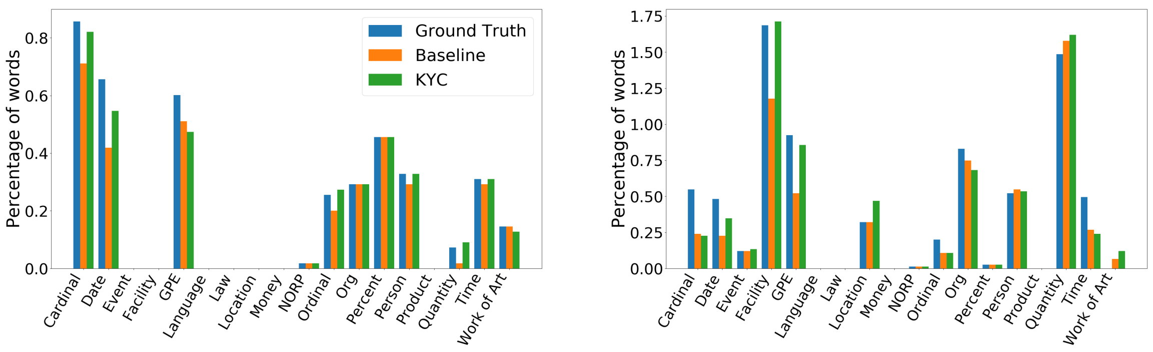

Figure 4: Proportion of true and predicted entity labels for different OOD clients (left) BC/Phoenix (right) BC/CCTV.

OOD Clients	OOD		ID	
	Base	KYC	Base	KYC
BC/CCTV + BC/Phoenix	63.8	75.3	86.0	79.3
BN/PRI + BN/VOA	88.7	90.5	84.5	78.7
NW/WSJ + NW/Xinhua	73.9	82.7	80.8	71.4
BC/CNN + TC/CH	78.3	80.3	85.6	79.9
WB/Eng + WB/a2e	76.2	76.4	86.4	79.6
Average	76.2	81.0	84.7	77.8

Table 8: Performance on the NER task on the Ontonotes dataset when using Binary Bag of Words as the client sketch.

OOD Clients	OOD		ID	
	Base	KYC	Base	KYC
BC/CCTV + BC/Phoenix	63.8	61.5	86.0	83.0
BN/PRI + BN/VOA	88.7	82.3	84.5	85.2
NW/WSJ + NW/Xinhua	73.9	82.3	80.8	75.0
BC/CNN + TC/CH	78.3	72.5	85.6	83.2
WB/Eng + WB/a2e	76.2	78.3	86.4	82.5
Average	76.2	75.4	84.7	81.8

Table 9: Performance on the NER task on the Ontonotes dataset when using sentence embddings averaged over an extracted summary.

OOD Clients	OOD			ID		
	Base	Sali-ence	Avg Len	Base	Sali-ence	Avg Len
Electronics+Games	86.4	88.1	86.9	88.5	89.0	88.6
Industrial+Tools	87.4	87.6	88.3	88.2	88.9	88.8
Books+Kindle Store	83.5	84.6	84.5	88.0	88.9	89.0
CDs+Digital Music	82.5	83.0	83.1	89.0	89.0	89.0
Arts+Automotive	89.9	90.6	90.2	88.2	88.6	88.5
Average	86.0	86.8	86.6	88.4	88.8	88.8

Table 10: Accuracy on the Sentiment Analysis task when using average review length as the client sketch. Columns "Saliency" and "Avg Len" refer to using KYC with the default saliency features and normalized review lengths as client sketches, respectively.

document when computing this vector. The corresponding numbers using this sketch are shown in Table 7 and are only slightly worse than the salience features.

- **Binary Bag of Words (BBoW)**: A binary vector of the same size as vocabulary is assigned to each client while setting the bit corresponding to a word on if the word has occurred in the client's data. We notice an improvement on the OOD set but a significant drop in ID numbers as seen in Table 8, 6. We attribute this to the strictly low representative power of BBoW sketches compared to the other sketches. The available train data for NER is laced with rogue clients which are not labeled and are instead assigned the default tag: "O". Proportion of KYC's improvement on this task comes from the ability to distinguish bad clients and keeping their parameters from not affecting other clients. This, however, is not possible when the representative capacity of the sketch is compromised. Thereby we do worse on ID using this sketch but not on OOD meaning the model does worse on the bad clients (which are only part of ID, and not OOD).
- **Contextualized Embedding of Summary**: We also experiment with using deep-learning based techniques to extract the topic and style of a client by using the "pooled" BERT embeddings averaged over sentences from the client. Since the large number of sentences from every client would lead to most useful signals being killed upon averaging, we first use a Summary Extractor (Barrios et al., 2016) to extract roughly 10 sentences per client and average the sentence embeddings over these sentences only. This method turns out to be ineffective in comparison to the other client sketches, indicating that sentence embeddings do not capture all the word-distribution

information needed to extract useful correction.

- **Average Instance Length:** For the task of Sentiment Analysis, we also experiment with passing a single scalar indicating average review length as the client sketch in order to better understand and quantify the importance of average review length on the performance of KYC. We linearly scale the average lengths so that all train clients have values in the range $[-1, 1]$. As can be seen in Table 10, this leads to a significant improvement over the baseline. In particular, the OOD splits CDs + Digital Music and Books + Kindle Store have reviews that are longer than the average and consequently result in improvements when augmented with average length information. The gains from review length alone are not higher than our default term-saliency sketch indicating that term frequency captures other meaningful properties as well.

D Results with Different Model Architectures

In this section we provide results for the different network architecture choices described in Section 3

- **Deep**: The architecture used is identical to that shown in Figure 1 barring $\bigoplus$, which now consists of an additional 128-dimensional non-linear layer before the final softmax transform Y_θ.
- **Decompose**: The final softmax layers is decomposed in to two. A scalar α is predicted from the client sketch using G_ϕ similar to KYC. The final softmax layer then is obtained through convex combination of the two softmax layers using α. Figure 5 shows the overview of the architecture.
- **MoE-g**: We use the client sketch as the drop-in replacement for encoded instance representation employed in Guo et al. (2018). The architecture is sketched in Figure 6. As shown in Table 13, this method works better than the standard MoE model, but worse than KYC.

OOD Clients	OOD		ID	
	Base	KYC	Base	KYC
BC/CCTV + BC/Phoenix	64.8	74.5	85.6	86.8
BN/PRI + BN/VOA	89.5	90.0	84.1	85.6
NW/WSJ + NW/Xinhua	74.4	80.6	80.2	92.8
BC/CNN + TC/CH	78.0	79.6	86.1	87.5
WB/Eng + WB/a2e	75.6	79.9	85.8	87.1
Average	76.5	80.9	84.4	86.0

Table 11: Performance on the NER task on the Ontonotes dataset using KYC-Deep.

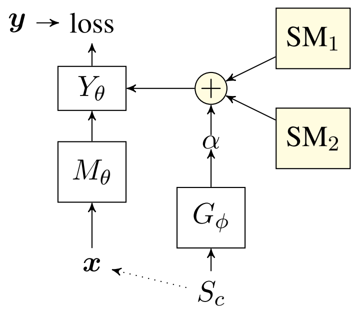

Figure 5: Decompose overview: $\bigoplus$ indicates a weighted linear combination. SM$_i$, $i \in \{1, 2\}$ represent the softmax matrices which are combined using weights α.

OOD Clients	OOD		ID	
	Base	KYC	Base	KYC
BC/CCTV + BC/Phoenix	64.1	56.0	85.6	86.3
BN/PRI + BN/VOA	89.6	89.9	84.6	85.5
NW/WSJ + NW/Xinhua	72.3	68.2	81.2	80.0
BC/CNN + TC/CH	78.5	77.5	85.9	86.6
WB/Eng + WB/a2e	75.5	71.0	86.1	86.7
Average	76.0	72.5	84.7	85.2

Table 12: Performance on the NER task on the Ontonotes dataset using Decompose.

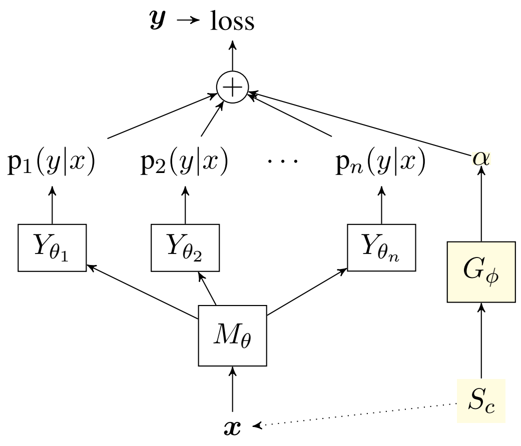

Figure 6: MoE-$\boldsymbol{g}$ overview: $\bigoplus$ indicates a weighted linear combination. $\mathrm{p}_i(y|x)$ represents the i^{th} expert's predictions and α represents weights for expert gating.

OOD Clients	OOD		ID	
	Base	KYC	Base	KYC
BC/CCTV + BC/Phoenix	64.8	74.7	85.6	84.0
BN/PRI + BN/VOA	89.5	88.3	84.1	83.6
NW/WSJ + NW/Xinhua	74.4	61.6	80.2	64.8
BC/CNN + TC/CH	78.0	73.7	86.1	82.1
WB/Eng + WB/a2e	75.6	76.3	85.8	84.4
Average	76.5	74.9	84.4	79.8

Table 13: Performance on the NER task on the Ontonotes dataset using MoE-$\boldsymbol{g}$.