Language Scaling for Universal Suggested Replies Model

Qianlan Ying[*1], Payal Bajaj[*2], Budhaditya Deb[2], Yu Yang[1], Wei Wang[†3],
Bojia Lin[1], Milad Shokouhi[2], Xia Song[2], Yang Yang[1], and Daxin Jiang[1]

[1]Microsoft, Beijing, China
[2]Microsoft, Bellevue, Washington, USA
[3]Qualtrics, Seattle, Washington, USA
`{qiying,Payal.Bajaj,Budha.Deb,yanyu}@microsoft.com`
`{bojial,milads,xiaso,yayan,djiang}@microsoft.com`
`tskatom@gmail.com`

Abstract

We consider the problem of scaling automated suggested replies for Outlook email system to multiple languages. Faced with increased compute requirements and low resources for language expansion, we build a single *universal* model for improving the quality and reducing run-time costs of our production system. However, restricted data movement across regional centers prevents joint training across languages. To this end, we propose a multitask *continual learning* framework, with auxiliary tasks and language adapters to learn universal language representation across regions. The experimental results show positive crosslingual transfer across languages while reducing *catastrophic forgetting* across regions. Our online results on real user traffic show significant gains in CTR and characters saved, as well as 65% training cost reduction compared with per-language models. As a consequence, we have scaled the feature in multiple languages including low-resource markets.

1 Introduction

Automated suggested replies or smart replies (SR) assist users to quickly respond with a short, generic, and relevant response, without users having to type in the reply. SR is an increasingly popular feature in many commercial applications such as Gmail, Outlook, Skype, Facebook Messenger, Microsoft Teams, and Uber (Kannan et al., 2016; Henderson et al., 2017a; Shang et al., 2015; Deb et al., 2019; Yue Weng, 2019). While the initial versions of this feature mostly targeted English users, making it available in multiple languages and markets is important not only from the perspective of product expansion but also from a linguistic inclusivity point of view.

In this paper we consider the problem of rapid scaling of the SR feature to multiple languages for Outlook. To develop such a system at production scale, we are faced with the following challenges.

- **Model management**: Language scaling increases the effort of training, deploying, and managing per-language models, which needs to be replicated for each language. In addition, one model per language increases the storage and compute requirements for the production servers, which can increase costs and occurrences of run-time issues.

- **Data constraints**: Developing models at production quality requires considerable effort in data collection and management. Due to regional market share and infrastructure constraints, rich and domain-specific data may not be available for all languages.

- **Data privacy and security policies**: Regional policies enforce data to be located in corresponding regions. For example, Spanish and Portuguese data are stored in North American (NAM) clusters while French data is stored in European (EUR) clusters. Data movement across regions is not allowed and this prevents leveraging commonly used multi-lingual co-training methods which require all the data stored to be in the same place.

To reduce the cost of model management, we propose to build a single *universal* SR model, capable of serving multiple languages and markets. To overcome data constraints, we propose to use *augmentation* with machine-translated (MT) data for languages without supervised data. To overcome privacy constraints, we propose a *continual* learning framework, where the model is trained sequentially across regions. To alleviate *catastrophic forgetting* (French, 1999; McCloskey and Cohen, 1989) in the continual learning process, we reinforce the universal properties via multi-task learning approach with public task-agnostic data, and an adapter-based model architecture that leverages domain-specific SR data and MT data.

Our experimental results followed with improvements shown on real user traffic illustrate the ef-

---

*Both authors contributed equally in the paper.
†Work performed at Microsoft Research.

fectiveness of the approach. As a consequence, we have rapidly scaled the feature in several languages including low-resource markets. Multilingual training for universal models is often very tricky to work in practice (especially with our data constraints). Thus, we demonstrate a significant accomplishment of a multi-lingual SR system running at production scale on millions of users, which saves resources while improving performance.

2 Core SR Model

The SR feature is similar to open-domain chatbots and task-oriented conversational agents, (Zhou et al., 2020; Henderson et al., 2019b; Fadhil and Schiavo, 2019; Xu et al., 2017; Okuda and Shoda, 2018; Kopp et al., 2018). In terms of usage, SR is closer to the latter, in that it assists users to complete a reply, instead of continuing an open-ended dialog. Following commonly used IR-based models in commercial SR applications (Henderson et al., 2017b; Deb et al., 2019), we use a dual encoder matching model for our SR system.

The matching model has two parallel encoders projecting input message and corresponding reply into a common representation space. Different encoders such as feed-forward and BiLSTM layers can be used here (Henderson et al., 2017a; Deb et al., 2019). More recently, (Devlin et al., 2018; Liu et al., 2019; Yang et al., 2019; Henderson et al., 2019a,b) show considerable improvements with transformer-based pre-trained models. Our English SR model uses a BERT equivalent (Devlin et al., 2018) encoder, while our mono-lingual baselines in other languages use BiLSTM encoders.

The model is trained on one-on-one message-reply (m-r) pairs from commercial email data. We minimize the symmetric loss function. It is a modified softmax on dot products between m-r encoding in equation 1 where $s_{i,j} = e^{\phi(m_i) \cdot \phi(r_j)}$. As described in (Deb et al., 2019), it was shown to improve the relevance by targeting at bi-directional *conversational* constraints.

$$p(m_i, r_i) = \frac{s_{i,i}}{\sum_j s_{i,j} + \sum_k s_{k,i} - s_{i,i}} \quad (1)$$

IR-based model requires a fixed response set. To generate that, we collect differentially private (DP) (Gopi et al., 2020) and anonymized replies, filtered for sensitive content from the training data which preserves user privacy while mining actual user responses. Furthermore, we use human curation
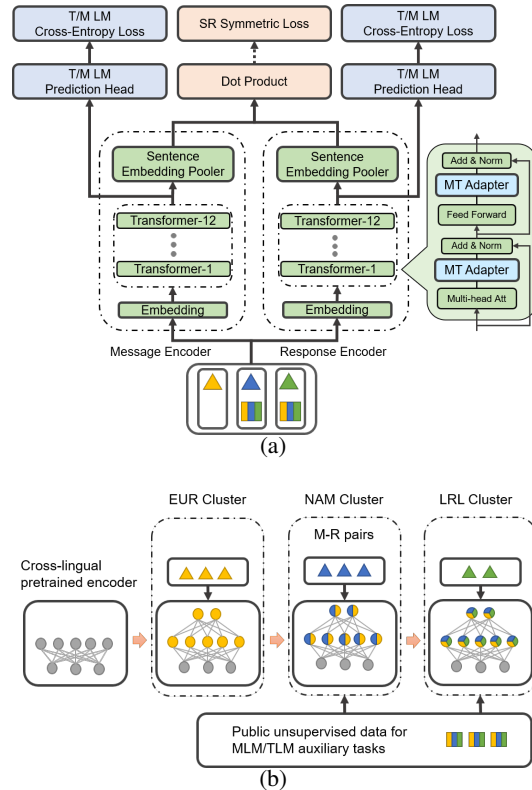

Figure 1: (a) Matching model architecture with symmetric loss and TLM/MLM cross-entropy loss. (b) Multi-task continual training loop for EUR->NAM->LRL clusters.

to edit responses for cultural-sensitivity, gender-neutrality, etc. DP filtration requires a large amount of data due to low yields. For low-resource markets, we translate English responses with human curation for cultural adaptation to languages and locales.

During prediction, we compute the matching score ($\cdot$) between the message and pre-computed response set vectors. Similar to (Henderson et al., 2017a; Deb et al., 2019), we add a language-model (LM) penalty representing the popularity of responses to bias the predictions towards more common ones. Translated responses inherit the penalty score from the corresponding English responses. Using this score in equation 2 we first select top N_1 responses, and down-select to top N_2 after deduplication using lexical clustering, before presenting to users.

$$Score = \phi(m_i) \cdot \phi_K(r_k)) + \alpha LM_K(r_k) \quad (2)$$

3 Universal SR Model

The universal SR model consists of parallel encoder architecture trained using symmetric loss function

similar to the core SR model. We initialize the m-r encoders with InfoXLM (Chi et al., 2020), an XLM-Roberta (Conneau et al., 2019) equivalent multi-lingual model as shown in as Figure 1(a) which creates language-agnostic text representation across 100 languages. The encoder is pre-trained with both publicly available and internal proprietary corpora and has shown good cross-lingual transfer capabilities on benchmarks such as XNLI (Conneau et al., 2018).

Using a universal pre-trained model in itself enables language expansion. However, as we discuss next, data movement constraints made training the universal model tricky, with performance frequently worse than single mono-lingual models.

3.1 Continual Learning

Joint training of universal encoders has led to enormous progress on standard benchmarks and industrial applications such as (Ranasinghe and Zampieri, 2020; Gencoglu, 2020).

However, privacy policies restrict the data movement across geographic clusters. This prevents the joint training at a single compute cluster. As a result, we train the model sequentially in a continual learning fashion by fine-tuning the model in one region, and then continue training in another.

The actual sequence of how this is conducted is important. We observed that keeping English at the last stage provides the best performance. This is likely because English data (which frequently contains bilingual data through code-switching) covers a large proportion in pre-training corpora, thus serving as an anchor in subsequent training stage to maintain the universal properties of the model.

3.2 Multi-task Learning

Training the SR model in multiple stages can lead to catastrophic forgetting, where new knowledge easily supplants old knowledge. This problem can be alleviated to some extent by freezing layers of the pre-trained encoders but is still significant after the model is fine-tuned with large corpora.

Several papers have leveraged self-supervised pre-training tasks based on bi-lingual parallel corpora to create or enhance cross-lingual representations (Devlin et al., 2018; Conneau et al., 2019; Chi et al., 2020). Following such approaches, we experiment with Translation Language Model (TLM) (Lample and Conneau, 2019) in continual learning to preserve the universal properties of the model.

A total of $79M$ translation pairs from WikiMatrix (Schwenk et al., 2019) and MultiParaCrawl (Aulamo et al., 2020) data including the languages considered in production are extracted as training data. In addition, we conduct an ablation study on auxiliary task selection by comparing with Masked Language Model (MLM) (Devlin et al., 2018) trained on $370M$ samples from Wikipedia.

The multi-task training alternates between SR and auxiliary tasks according to a set proportion of mini-batches in an epoch. The proportion controls the trade-offs between the tasks, to achieve the desired levels of performance in the system.

3.3 Data Augmentation

Native supervised data (m-r pairs) is currently not available for low-resource languages. In such cases, English data is leveraged to generate pseudo m-r pairs using machine-translation (MT). We utilize MT data in continual learning process with auxiliary tasks, or with adapters (Houlsby et al., 2019) by introducing additional parameters in the transformer layers. When training with adapters, we freeze all parameters except the adapters.

3.4 Universal Model Training Loop

The production system targets 5 high-resource languages (HRL): Spanish (ES), Portuguese (PT), French (FR), German (DE), Italian (IT) with rich native data, and 5 low-resource languages (LRL): Chinese (ZH), Japanese (JA), Dutch (NL), Czech (CS) and Hungarian (HU) without any supervised data. English (EN) serves as pivot language in our experiments. As shown in Table 1, the data is distributed across Europe (EUR), North America (NAM) and a dedicated cluster storing MT data for LRL. Data movement across these regions is not allowed. Public task-agnostic data for auxiliary tasks in 8 languages is accessible in all regions.

Region	Languages	Category
EUR	DE, IT, FR	High-resource
NAM	ES, PT, EN	High-resource
LRL	ZH, JA, NL, CS, HU	Low-resource*

Table 1: Regional distribution of training data for different languages. *: data translated from EN.

We train the model sequentially in 3 stages as shown in Figure 1(b). First, we jointly train the model in EUR for FR, DE, and IT. Next, we move the model to NAM and continue train with EN, ES,

and PT along with auxiliary task. Finally, in LRL, we train the model on machine translated m-r pairs along with original EN data in 2 different ways: (1) jointly train with auxiliary task, or (2) infuse the model with low-resource language adapters. In all stages, we freeze the embedding layer of the encoder during fine-tuning. According to previous studies (Lee et al., 2019; Peters et al., 2019), freezing partial layers can maintain the model quality while reducing training time during fine-tuning. We observed that freezing embedding layer provides a good balance between micro-batch size per GPU (low if no layers are frozen) and learning capacity of the model (low if many layers are frozen).

3.5 Universal Model Graph for Serving

For deployment, we create a composite graph with pre-computed response vectors of all languages embedded into the main model. A separate language identifier switches the prediction vectors to the predicted language of the input at run-time. Besides, several auxiliary models are added in online system to decide whether to trigger the universal model according to the characteristics of input message such as length and detected language.

4 Experiments and Results

The training data is collected and processed without any eyes access from commercial users in Outlook email system. To be more specific, we filter 50M m-r pairs from one-to-one conversations for each high-resource language, and translate 20M m-r pairs for each low-resource language. Considering the m-r length distribution, we truncate m-r pairs to (96, 64) tokens as training data, and filter out messages longer than 96 tokens during inference, so that the model is more focused on providing quick responses to short messages. The response set size for each language is 20K, filtered or trans-created from English native data.

In all three stages of training, we use an effective batch size of 16K. We utilize the Adam optimizer (Kingma and Ba, 2014) with weight decay and set peak learning rates as [5e-4, 3e-4, 1e-4] for three stages respectively. We train up to 30 epochs from which the best model is selected based on validation set loss over all languages.

For MLM/TLM objectives, we use single-token masking, the task proportion is set as 0.5. The final loss of the model is sum of symmetric loss and auxiliary task loss. For adapters, we use the

hidden dimension of 256 in the bottleneck architecture and initialize these parameters with a normal distribution of mean 0 and standard deviation 0.01. According to our observation, high standard deviation for initialization can cause divergence. All experiments are conducted with 16 Nvidia V100-32GB GPU cards.

During prediction, we pick top $N_1 = 30$ responses according to equation 2, and then cluster the ranked results and down-select $N_2 = 3$ responses as final prediction.

4.1 Offline Evaluation Metrics and Sets

We compute evaluation metrics based on two kinds of evaluation sets. The first test set samples m-r pairs, where reply is contained in the response set (GoldenMR) and is used for computing the ranking metric, Mean Reciprocal Rank: $MRR = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{Rank_i}$, for the top 15 predictions.

The second set consists of general m-r pairs (GenMR) where the reply is not restricted to the response set. *weighted*-ROUGE metrics is computed on final 3 responses with the reference response over uni/bi/tri-grams ($W\_ROUGE = \sum_{i=1}^{3} \frac{1}{w_i} ROUGE_i(Ref, Rep_k)$), with weights of $1:2:3$ proportions.

We use ~50K GoldenMR and 500K GenMR dataset for each language. For languages without native data, an evaluation proxy with MT data is used for model selection before online deployment. We give a higher preference to ROUGE as it showed higher correlation to our online metrics.

4.2 Online Evaluation Metrics

For the deployed models in production, we measure the following online metrics on real user traffic.

Click-through rate (CTR): the ratio of the count of replied emails with SR clicks over all emails that the feature is rendered.

Usage: the ratio of count of replied emails with SR clicks to all replied emails. This captures the contribution of SR to all Email replies.

Char-saved: the average number of characters-saved by clicking the selected reply.

4.3 Results

The model is evaluated on the international markets we are expanding to. English is excluded as EN model is well established. Results on baseline (existing per-language production models) and universal models for high-resource markets are reported in Table 2. Results targeting new markets without

any native data are reported in Table 3. Entries in the tables are defined as follows:

BiLSTM: Per-language (mono-lingual) production models for non-EN markets as the baseline and also the control setting of online A/B tests. Here the encoders have shared embedding size of 320 and 2 BiLSTM layers with hidden size of 300.

UniPLM-[*NAM/EUR*]: Universal model created by fine-tuning pre-trained multi-lingual encoders for EUR and NAM regions respectively.

UniPLM-HRL: The model across the first 2 stages with the universal training loop in Figure 1(b). In the second stage, the model is fine-tuned along with TLM auxiliary task with multi-lingual unsupervised data. This is the first universal model candidate that breaks down the data boundary across High-Resource Languages (HRL).

Reg	Lang	Model	MRR	W_ROUGE
EUR	DE	BiLSTM-de	0.3263	0.0685
		UniPLM-EUR	**0.4185**	**0.0698**
		UniPLM-HRL	0.3323	0.0663
	FR	BiLSTM-fr	0.4569	0.0642
		UniPLM-EUR	**0.4721**	**0.0647**
		UniPLM-HRL	0.4135	0.0624
	IT	BiLSTM-it	0.3300	0.0330
		UniPLM-EUR	**0.4819**	**0.0385**
		UniPLM-HRL	0.4186	0.0360
NAM	ES	BiLSTM-es	0.3248	0.0511
		UniPLM-NAM	0.3186	**0.0565**
		UniPLM-HRL	**0.3319**	0.0552
	PT	BiLSTM-pt	**0.4383**	0.0552
		UniPLM-NAM	0.4216	**0.0577**
		UniPLM-HRL	0.4154	0.0563

Table 2: Evaluation on HRL (EUR and NAM) with UniPLM-HRL via continual multi-task learning and production baselines. The best results are in bold.

For new languages without native data, we continue to train the base universal model (UniPLM-HRL) with MT data with two approaches.

UniPLM-All-CL: The UniPLM-HRL model exported to LRL region trained with MT data (and native EN data) with SR and TLM multi-task objectives.

UniPLM-All-ADP: The model trained with MT-adapter, with all parameters frozen except for adapters parameters.

4.4 Model Quality Analysis

Table 2 compares the universal model UniPLM-HRL with both per-language baselines and per-region models. Table 3 shows the results with the low-resource languages, which are trained with

Reg	Lang	Model	MRR	W_ROUGE
EUR	DE	UniPLM-HRL	**0.3323**	0.0663
		UniPLM-All-CL	0.3103	**0.0686**
	FR	UniPLM-HRL	0.4135	0.0624
		UniPLM-All-CL	**0.4207**	**0.0659**
	IT	UniPLM-HRL	0.4186	0.0360
		UniPLM-All-CL	**0.4274**	**0.0374**
NAM	ES	UniPLM-HRL	**0.3319**	**0.0552**
		UniPLM-All-CL	0.3160	0.0551
	PT	UniPLM-HRL	**0.4154**	**0.0563**
		UniPLM-All-CL	0.3783	0.0561
LRL	ZH	UniPLM-HRL	0.1365	0.0740
		UniPLM-All-CL	0.2638	0.0869
		UniPLM-All-ADP	**0.3024**	**0.0901**
	JA	UniPLM-HRL	0.1475	0.1010
		UniPLM-All-CL	0.3281	0.1106
		UniPLM-All-ADP	**0.3719**	**0.1180**
	NL	UniPLM-HRL	0.0638	0.0371
		UniPLM-All-CL	0.1822	0.0436
		UniPLM-All-ADP	**0.2490**	**0.0480**
	CS	UniPLM-HRL	0.0366	0.0386
		UniPLM-All-CL	0.1312	0.0441
		UniPLM-All-ADP	**0.2612**	**0.0526**
	HU	UniPLM-HRL	0.0420	0.0356
		UniPLM-All-CL	0.0779	0.0776
		UniPLM-All-ADP	**0.2615**	**0.0907**

Table 3: Results with UniPLM-All-CL and UniPLM-All-ADP continually augmented with MT data.

data augmentation approach involving MT data, with multi-task learning or adapters.

Per-language vs. Universal Model: The BiLSTM production models serve as strong baselines and have comparable MRR for UniPLM-NAM in ES and PT (Table 2). UniPLM-EUR has better performance than the BiLSTM production models. Overall, the Uni-PLM models have comparable or better performance than the monolingual baselines.

UniPLM-NAM/EUR vs. UniPLM-HRL: Table 2 also shows no appreciable difference in ROUGE metrics when training the model in 2 stages. In addition, the model outperforms BiL-STM per-language models on MRR on ES, DE, FR, and IT.

The above two comparisons show that for high-resource languages, we do not suffer significant degradation in quality with single stage and two-stage universal models.

Performance on LRL: Table 3 compares the UniPLM-All-CL and UniPLM-All-ADP with UniPLM-HRL model on low-resource languages. While UniPLM-HRL shows poor ranking performance, UniPLM-All-CL significantly improves on all metrics for LRL, while preserving the ROUGE performance on the other 5 languages. With

adapters, UniPLM-All-ADP outperforms other models on all metrics in low-resource languages while keeping the performance unchanged (as a result of freezing the UniPLM-HRL model) in both EUR and NAM.

Overall, the results demonstrate the effectiveness of MT data augmentation in low-resource languages. We observe slight performance degradation on EUR and NAM languages caused by continual training on MT data. This may be due to imperfect translation. However we can mitigate these losses with MT-adapters which are quite promising as they increase the parameters by just 4.3% and even improves training efficiency as we can freeze all other parameters during fine tuning.

Reg	Lang	Model	MRR	W_ROUGE
EUR	DE	UniPLM-HRL	0.3323	0.0663
		-TLM	**0.3643**	**0.0701**
		-TLM+MLM	0.3070	0.0596
	FR	UniPLM-HRL	**0.4135**	**0.0624**
		-TLM	0.3772	0.0583
		-TLM+MLM	0.4126	0.0606
	IT	UniPLM-HRL	0.4186	**0.0360**
		-TLM	**0.4284**	0.0359
		-TLM+MLM	0.4035	0.0343
NAM	ES	UniPLM-HRL	**0.3319**	**0.0552**
		-TLM	0.2958	0.0543
		-TLM+MLM	0.3023	0.0537
	PT	UniPLM-HRL	0.4154	**0.0563**
		-TLM	0.4176	0.0561
		-TLM+MLM	**0.4234**	0.0559

Table 4: Results with variations on UniPLM-HRL. -TLM denotes removing TLM and -TLM+MLM denotes replacing with MLM in continual learning.

Reg	Lang	Model	MRR	W_ROUGE
EUR	DE	UniPLM-HRL	0.3323	0.0663
		+EUR	0.4272	0.0708
	FR	UniPLM-HRL	0.4135	0.0624
		+EUR	0.4818	0.0660
	IT	UniPLM-HRL	0.4186	0.0360
		+EUR	0.4851	0.0388
NAM	ES	UniPLM-HRL	0.3319	0.0552
		+EUR	0.2125	0.0456
	PT	UniPLM-HRL	0.4154	0.0563
		+EUR	0.3298	0.0505

Table 5: Results with 2-stage and replay-based continual learning. +EUR denotes replaying UniPLM-HRL with EUR m-r pairs.

4.5 Ablation Studies

MLM and TLM auxiliary tasks: Table 4 investigates contributions of auxiliary tasks in UniPLM-

HRL model. We remove TLM objective as *-TLM* which represents continue training only on SR task, and replace TLM with MLM objective as *-TLM+MLM* which represents joint training with SR and MLM tasks. UniPLM-HRL with TLM task shows improvements over MLM task and also outperforms single SR task for W_ROUGE for all languages except DE. We hypothesize that TLM uses bi-lingual corpora which helps align representations for semantically similar text from different languages in task-specific fine-tuning. Furthermore, TLM objective can be interpreted as maximizing mutual information between cross-lingual contexts implicitly (Chi et al., 2020). It demonstrates that such inductive biases in auxiliary tasks are important for cross-lingual transfer in universal models.

Replay in continual learning: We continue to train the UniPLM-HRL model by rehearsing the old data in EUR as *+EUR*. In Table 5, *+EUR* we see severe regression on NAM languages, despite the improvement on EUR languages. The *replay* concept in continual learning (McClelland, 1998) fails here due to the two reasons. First, forgetting is the quintessential mode of continual learning. Second, EUR iteration doesn't contain the pivot language English training data. Continual learning requires delicately maintaining the universal properties through knowledge anchors which is difficult to achieve in practice.

4.6 Online Results

Based on the offline metrics, we selected UniPLM-HRL as the first candidate for online tests in our production system. Using BiLSTM per-language model as the control, we conducted a 2-week A/B test with 5% user traffic for each model per language/region. Table 6 presents the results for different languages. We observe statistically significant gain in ES (CTR) and FR (Char-saved). While there are regressions in other languages, they are not statistically significant ($p > 0.5$)

Lang	CTR (p-val)	Usage (p-val)	Char-saved (p-val)
ES	**4.20%** (0.0163)	**7.71%** (0.0001)	-0.91% (0.6592)
PT	0.00% (0.3690)	0.00% (0.3264)	3.32% (0.2737)
FR	-3.51% (0.1636)	-3.14% (0.2773)	**5.08%** (0.0495)
IT	-3.62% (0.4506)	-7.59% (0.1515)	7.03% (0.0602)
DE	2.80% (0.5147)	-1.07% (0.8233)	5.39% (0.1193)

Table 6: Online metrics for UniPLM-HRL model. The control model is BiLSTM in each language. The numbers with p-val < 0.05 are in bold.

Overall, the universal model is generally better

or at par compared to their mono-lingual baselines. This has allowed us to deploy the universal model to 100% of users in the 5 languages. An extended universal model supporting low-resource languages is getting deployed during the writing of this paper.

Compared with per-language separate model building, the effort of model training, inference stack and deployment can be substantially reduced, though the process of training data and response collection, and human evaluation for all our targeted languages are still required. Overall, around 65% training and performance improvement time cost can be saved with one single universal model target at 5 languages. We expect even higher amortized serving costs reductions as the approach is scaled to more languages.

5 Conclusions

This paper presents our approach of scaling automated suggested replies with one universal model. Faced with compute resource and data privacy constraints, we propose a multi-task continual learning framework with auxiliary tasks, and data augmentation with adapter-based model architecture. The universal model in production saves significant compute resources and model management overhead, while allowing us to train across regional data boundaries. In addition, the process allows us to *cold-start* in new markets even when no supervised data exists. Based on the promising offline and online results, we have deployed the model in several languages and plan to extend the process for 20 languages around the world.

References

Mikko Aulamo, Umut Sulubacak, Sami Virpioja, and Jörg Tiedemann. 2020. Opustools and parallel corpus diagnostics. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 3782–3789.

Zewen Chi, Li Dong, Furu Wei, Nan Yang, Saksham Singhal, Wenhui Wang, Xia Song, Xian-Ling Mao, Heyan Huang, and Ming Zhou. 2020. Infoxlm: An information-theoretic framework for cross-lingual language model pre-training. *arXiv preprint arXiv:2007.07834*.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*.

Alexis Conneau, Guillaume Lample, Ruty Rinott, Adina Williams, Samuel R Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. Xnli: Evaluating cross-lingual sentence representations. *arXiv preprint arXiv:1809.05053*.

Budhaditya Deb, P. Bailey, and M. Shokouhi. 2019. Diversifying reply suggestions using a matching-conditional variational autoencoder. In *NAACL-HLT*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Ahmed Fadhil and Gianluca Schiavo. 2019. Designing for health chatbots. *arXiv preprint arXiv:1902.09022*.

Robert M French. 1999. Catastrophic forgetting in connectionist networks. *Trends in cognitive sciences*, 3(4):128–135.

Oguzhan Gencoglu. 2020. Large-scale, language-agnostic discourse classification of tweets during covid-19. *Machine Learning and Knowledge Extraction*, 2(4):603–616.

Sivakanth Gopi, Pankaj Gulhane, Janardhan Kulkarni, Judy Hanwen Shen, Milad Shokouhi, and Sergey Yekhanin. 2020. Differentially private set union. *arXiv preprint arXiv:2002.09745*.

Matthew Henderson, Rami Al-Rfou', Brian Strope, Yun-Hsuan Sung, László Lukács, Ruiqi Guo, Sanjiv Kumar, Balint Miklos, and Ray Kurzweil. 2017a. Efficient Natural Language Response Suggestion for Smart Reply. *CoRR*, abs/1705.00652.

Matthew Henderson, Rami Al-Rfou, Brian Strope, Yun-Hsuan Sung, László Lukács, Ruiqi Guo, Sanjiv Kumar, Balint Miklos, and Ray Kurzweil. 2017b. Efficient natural language response suggestion for smart reply. *arXiv preprint arXiv:1705.00652*.

Matthew Henderson, Inigo Casanueva, Nikola Mrkšić, Pei-Hao Su, Tsung-Hsien Wen, and Ivan Vulić. 2019a. Convert: Efficient and accurate conversational representations from transformers. *arXiv preprint arXiv:1911.03688*.

Matthew Henderson, Ivan Vuli'c, Daniela Gerz, Iñigo Casanueva, Paweł Budzianowski, Sam Coope, Georgios Spithourakis, Tsung-Hsien Wen, Nikola Mrksi'c, and Pei-Hao Su. 2019b. Training neural response selection for task-oriented dialogue systems. In *Proceedings of ACL*.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for NLP. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2790–2799. PMLR.

Anjuli Kannan, Karol Kurach, Sujith Ravi, Tobias Kaufmann, Andrew Tomkins, Balint Miklos, Gregory S. Corrado, László Lukács, Marina Ganea, Peter Young, and Vivek Ramavajjala. 2016. Smart Reply: Automated Response Suggestion for Email. In *KDD*.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Stefan Kopp, Mara Brandt, Hendrik Buschmeier, Katharina Cyra, Farina Freigang, Nicole Krämer, Franz Kummert, Christiane Opfermann, Karola Pitsch, Lars Schillingmann, et al. 2018. Conversational assistants for elderly users–the importance of socially cooperative dialogue. In *Proceedings of the AAMAS Workshop on Intelligent Conversation Agents in Home and Geriatric Care Applications co-located with the Federated AI Meeting*, volume 2338.

Guillaume Lample and Alexis Conneau. 2019. Cross-lingual language model pretraining. *arXiv preprint arXiv:1901.07291*.

Jaejun Lee, Raphael Tang, and Jimmy Lin. 2019. What would elsa do? freezing layers during transformer fine-tuning. *arXiv preprint arXiv:1911.03090*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

James L McClelland. 1998. Complementary learning systems in the brain. a connectionist approach to explicit and implicit cognition and memory. *Annals of the New York Academy of Sciences*, 843:153.

Michael McCloskey and Neal J Cohen. 1989. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24, pages 109–165. Elsevier.

Takuma Okuda and Sanae Shoda. 2018. Ai-based chatbot service for financial industry. *Fujitsu Scientific and Technical Journal*, 54(2):4–8.

Matthew E Peters, Sebastian Ruder, and Noah A Smith. 2019. To tune or not to tune? adapting pretrained representations to diverse tasks. *arXiv preprint arXiv:1903.05987*.

Tharindu Ranasinghe and Marcos Zampieri. 2020. Multilingual offensive language identification with cross-lingual embeddings. *arXiv preprint arXiv:2010.05324*.

Holger Schwenk, Vishrav Chaudhary, Shuo Sun, Hongyu Gong, and Francisco Guzmán. 2019. Wikimatrix: Mining 135m parallel sentences in 1620 language pairs from wikipedia. *arXiv preprint arXiv:1907.05791*.

Lifeng Shang, Zhengdong Lu, and Hang Li. 2015. Neural responding machine for short-text conversation. *arXiv preprint arXiv:1503.02364*.

Anbang Xu, Zhe Liu, Yufan Guo, Vibha Sinha, and Rama Akkiraju. 2017. A new chatbot for customer service on social media. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pages 3506–3510.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in neural information processing systems*, pages 5753–5763.

Franziska Bell Gokhan Tur Yue Weng, Huaixiu Zheng. 2019. Occ: A smart reply system for efficient in-app communications. *arXiv preprint arXiv:1907.08167*.

Li Zhou, Jianfeng Gao, Di Li, and Heung-Yeung Shum. 2020. The design and implementation of xiaoice, an empathetic social chatbot. *Computational Linguistics*, 46(1):53–93.