A Semi-supervised Multi-task Learning Approach to Classify Customer Contact Intents

Li Dong
Amazon
ldonga@amazon.com

Matthew C. Spencer
Amazon
matsp@amazon.com

Amir Biagi
Amazon
biagiab@amazon.com

Abstract

In the area of customer support, understanding customers' intents is a crucial step. Machine learning plays a vital role in this type of intent classification. In reality, it is typical to collect confirmation from customer support representatives (CSRs) regarding the intent prediction, though it can unnecessarily incur prohibitive cost to ask CSRs to assign existing or new intents to the mis-classified cases. Apart from the confirmed cases with and without intent labels, there can be a number of cases with no human curation. This data composition (Positives + Unlabeled + multiclass Negatives) creates unique challenges for model development. In response to that, we propose a semi-supervised multi-task learning paradigm. In this manuscript, we share our experience in building text-based intent classification models for a customer support service on an E-commerce website. We improve the performance significantly by evolving the model from multiclass classification to semi-supervised multi-task learning by leveraging the negative cases, domain- and task-adaptively pretrained ALBERT on customer contact texts, and a number of un-curated data with no labels. In the evaluation, the final model boosts the average AUC ROC by almost 20 points compared to the baseline finetuned multiclass classification ALBERT model.

1 Introduction

As machine learning makes rapid advances in the area of natural language processing (NLP), it is becoming more common to aid customer support representatives (CSRs) with NLP models. This not only ensures timely and consistent replies to customers, but also reduces operational costs for organizations. We can see successful use cases from organizations such as Alibaba (Fu et al., 2020), Uber (Molino et al., 2018), Square (Fotso et al., 2018), AT&T (Gupta et al., 2010), IBM (Mani et al., 2018), Los Alamos National Laboratory (DeLucia and Moore, 2020), and US Navy (Powell et al., 2020). In general, identifying the intents of the coming contacts is the first step in customer support. Therefore, accurate intent classification is crucial.

Intent classification is a broad topic mostly falling under the umbrella of NLP. In this manuscript, we limit our discussion to intent classification in the area of customer support. In the past two decades, researchers have been trying to improve the efficiency of customer support by detecting customer intents with machine learning approaches (Molino et al., 2018; Powell et al., 2020; DeLucia and Moore, 2020; Hui and Jha, 2000; Gupta et al., 2010; Fotso et al., 2018; Mani et al., 2018; Sarikaya et al., 2011; Gupta et al., 2006; Xu and Sarikaya, 2013). We can loosely categorize these approaches into text classification (Molino et al., 2018; Powell et al., 2020; DeLucia and Moore, 2020; Hui and Jha, 2000; Gupta et al., 2010; Fotso et al., 2018), question-answer (QA) system (Mani et al., 2018) and automatic speech recognition (ASR) (Sarikaya et al., 2011; Gupta et al., 2006; Xu and Sarikaya, 2013). In this manuscript, we focus on using text classification methods to classify intents for customer support. To deal with unstructured text data, researchers use handcrafted features (Hui and Jha, 2000; Gupta et al., 2010), Bag-of-Words type of features (Powell et al., 2020), features from topic modeling (DeLucia and Moore, 2020) and vectorization type of features, such as word2vec (Fotso et al., 2018; Molino et al., 2018) and doc2vec (DeLucia and Moore, 2020). By consuming these features, classifiers determine the intent of a case and the case can be routed to specialists (Molino et al., 2018; Gupta et al., 2010; DeLucia and Moore, 2020; Powell et al., 2020) and/or a reply template from the "Answer Bank" can be provided (Molino et al., 2018; Fotso et al., 2018; Hui and Jha, 2000). A general intelligent

Proceedings of the 4th Workshop on e-Commerce and NLP (ECNLP 4), pages 49–57

customer support loop can be seen in Figure 1.

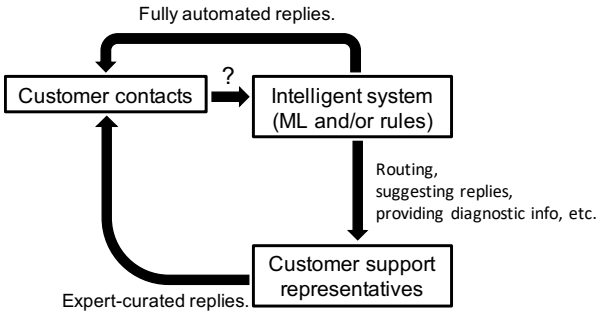

Figure 1: Intelligent customer support loop

To meet ever-changing business needs, the intent taxonomy is generally under active development (Molino et al., 2018; Fotso et al., 2018; DeLucia and Moore, 2020). It is not realistic to manually relabel all cases after each intent taxonomy update. This means that we have positive cases (P) with assigned intents and unlabeled cases (U) in data. Moreover, to maintain a high standard of customer satisfaction, intent classification is typically a human-in-the-loop process (Fu et al., 2020; Molino et al., 2018; Fotso et al., 2018; Gupta et al., 2010; Powell et al., 2020). Specifically, the CSRs are asked to confirm the intent predictions, a process we refer to as "curation" in this manuscript. The negative cases (N) identified by CSRs are indeed hard cases, since their prediction scores are above the preset confidence threshold yet they are mis-classified by the existing model. It is an active research area to create classifiers with only P and U (Elkan and Noto, 2008; Xu et al., 2017). Some research has explored models that also include N, but they have been only concerned with binary classifiers (Fei and Liu, 2015; Hsieh et al., 2019; Li et al., 2010).

In this manuscript, we adopt the semi-supervised paradigm and the multi-task approach to deal with the U and the multiclass N, respectively. Moreover, in contrast to the above-mentioned works about intent classification for customer support, we use the ALBERT pretrained language model (Lan et al., 2019) plus domain- and task-adaptive pretraining (Ramponi and Plank, 2020; Gururangan et al., 2020) to process texts. In the following sections, we describe how these techniques improve the model performance.

The paper outline is as follows. We start with Section 2 by elaborating the business background and how we pose it as a machine learning problem. Then we describe the details of the training data and models in Section 3, compare the models by conducting experiments with real data in Section 4, and discuss the results in Section 5. We conclude in Section 6.

2 Background

The E-commerce website of interest receives many support requests from customers in each second. There is a team of CSRs to actively address the requests via phone, online chat, and email channels. Identifying appropriate requests and grouping them into categories is not a trivial task. While a deep discussion of the taxonomy building process is out of the scope of this manuscript, it is sufficient to know that we have a taxonomy system that is similar to those described in (Molino et al., 2018; Fotso et al., 2018), where customized reply templates are pre-compiled for each customer contact intent. This study elaborates our journey building machine learning models to classify the intents.

3 Methodology

Since the inception of BERT (Devlin et al., 2019), an abundance of research in the area of NLP has demonstrated it to be an effective approach to transfer knowledge from pretrained language models to downstream tasks (Xia et al., 2020; Wang et al., 2018, 2019a; Rajpurkar et al., 2016; Lai et al., 2017). Following BERT's architecture, there is a stream of research that achieve comparable or better performance, to name a few (Lan et al., 2019; Liu et al., 2019; Wang et al., 2019b; Clark et al., 2020; Yang et al., 2019; Sanh et al., 2019). Among these BERT variants, ALBERT aims to strike a balance between model performance and model size (Lan et al., 2019). Therefore, we use albert-base-v2 as the backbone encoder and perform further pretraining and finetuning. The implementation is based on Transformers from Huggingface (Wolf et al., 2019).

3.1 Training Data

3.1.1 Features

The input to the model is a collection of emailed support requests in text format. The texts are minimally preprocessed, including removing invalid characters, lowercasing letters and replacing some obvious entities with consistent words, such as replacing urls and emails to url_id and email_id.

3.1.2 Targets

In industrial machine learning applications, it is typical to construct a feedback loop to collect training data. In most cases, it is straightforward to obtain a simple "yes" or "no" from human labelers with respect to the predictions. That means the human labelers only need to accept or dismiss the recommendations. Those "yes" cases are confirmed positive ones with explicit labels. However, in a common multiclass classification setting, the "no" cases can have any other label, so the labels are effectively unknown. In some scenarios, such as object detection in computer vision, it is not hard to ask human labelers to assign a label to those negative cases. However, it is highly non-trivial to ask for a valid label in many NLP applications, owing to the size of the taxonomy and the necessary domain-expertise, as is the case for the intent classification in this manuscript. Therefore, in our training data, we only have "yes" or "no" feedback to each case in each intent class.

Since the scope of the intent taxonomy is not trying to cover all customer support requests, there are many requests falling out of the scope of the taxonomy but still scored by the model. The negative cases are either out-of-scope requests or in-scope requests falling in the wrong bucket. The former one is more probable, since the requests are false positives for existing classes with high confidence scores above the preset thresholds. In this manuscript, we tried two ways to deal with this situation.

1. We can simply exclude the negative cases from training data, since they do not come with labels. In this scenario, it is a multiclass classification model trained on positive cases, i.e. confirmed intents. However, we lose valuable signals by excluding the negative cases.

2. Since the negative cases are indeed hard negatives and contain valuable signals, we can use the multi-task learning paradigm to elegantly treat the negatives for each intent class as the negative samples for a binary classification task. In this scenario, we have a binary classification task for each class plus a multiclass classification task for all classes. It is also not necessary to examine the negative cases and assign them to appropriate new or existing classes, especially when the labeling efforts outweigh the benefits it could bring to model development. With this approach, we make full use of the signals in the training data.

Apart from the multiclass positive (P) and negative (N) cases mentioned above, we also have the un-curated cases that do not come with labels, i.e. the U cases. We adopt an iterative semi-supervised approach to deal with them. The approach is described in Section 3.2.2.

3.2 Models

3.2.1 ALBERT

Following the pretraining-finetuning framework for language models, we start with a finetuned ALBERT. We simply remove the masked language model (MLM) head and the sentence order prediction (SOP) head from ALBERT and add a sequence classification head. Following the convention from (Devlin et al., 2019), the final hidden vector corresponding to the first input token [CLS] is used for classification. We denote this vector as the classification vector in the rest of the manuscript. We note that this ALBERT model is trained as a multiclass classification with only *positive* cases.

3.2.2 SS MT D/TAPT ALBERT

The pretrained language models are mostly trained on well-known corpora, such as Wikipedia, Common Crawl, BookCorpus, Reddit, etc. However, in many cases, we need to apply the language models to very different domains, like BioMed, scientific publication, or product reviews. For these types of problems, researchers have found that, in addition to finetuning on specific downstream tasks, it is beneficial to adapt the language models to the domain- and task-specific corpus, i.e. domain-adaptive pretraining (DAPT) and task-adaptive pretraining (TAPT) (Gururangan et al., 2020). This is achieved by further training the language modeling tasks, such as MLM, with the corpus of the domain and the task. We note that it can be difficult to rigorously define *domain* in NLP. For the DAPT training in this manuscript, we simply use customer contacts in the past few months as the domain corpus and follow the training recommendations from (Gururangan et al., 2020).

To make full use of the feedback from CSRs, we include the negatively confirmed cases and treat each class as a separate binary classification task in addition to the multiclass classification task. We accomplish the modeling with the multi-task (MT) learning paradigm (Liu et al., 2019). In this case,

we have $n+1$ tasks, i.e. n binary classification tasks and 1 multiclass classification task. As illustrated on the left of Figure 2, we train the model in an end-to-end fashion. This means the $n+1$ tasks are finetuned jointly sharing the same encoder. We note that every positive sample belongs to two tasks (the multiclass classification task and one binary task) and each negative sample only belongs to the corresponding binary classification task. In inferencing time, as illustrated on the right of Figure 2, the model first processes the case text through the encoder to get the classification vector. Then the multiclass classification task consumes the vector and predicts the class. In the end, the same vector is routed to the binary task corresponding to that class, predicting the probability of the intent class accepted by the CSRs.

To make it more concrete, we can see the training loss implementation in Equation (1). y^b is the binary label, i.e. 1 means it is a positive sample and its intent class is confirmed by CSRs with "yes". l^m is the multiclass task loss. $\boldsymbol{y}^m$ is the one-hot encoded n-dimensional multiclass label vector. $\boldsymbol{l}^b$ is the loss function vector for n binary tasks. N is the number of samples. Typical cross-entropy loss is used for all tasks here.

$$\mathcal{L} = \frac{1}{N}\sum_{i=1}^{N}(y_i^b \cdot l_i^m + \boldsymbol{y}_i^m \cdot \boldsymbol{l}_i^b) \tag{1}$$

For the inferencing process, we refer to Equations (2)-(5). $\boldsymbol{x}$ is the tokenized sequence vector. $\boldsymbol{u}$ is the classification vector, i.e. the embedding vector for the CLS token. f^m is the multiclass classifier. f_k^b is the binary classifier for intent class k.

$$\boldsymbol{u} = \mathit{Encoder}(\boldsymbol{x}) \tag{2}$$
$$\hat{\boldsymbol{y}}^m = f^m(\boldsymbol{u}) \tag{3}$$
$$k = \arg\max_i \hat{\boldsymbol{y}}^m(i), i \in [0...n-1] \tag{4}$$
$$\hat{y}^b = f_k^b(\boldsymbol{u}) \tag{5}$$

Moreover, we add the semi-supervised (SS) strategy to take advantage of the un-curated data. While a large volume of model predictions are reviewed by the CSRs each second, we believe there are still a number of qualified cases that we miss. Therefore, we can train the model, make prediction on the un-curated cases, choose the high-confidence ones, and re-train the model with the labeled data plus the high-confidence cases. We follow this in an iterative manner until the improvement diminishes such that it cannot justify the training cost. We note that we only augment the data of the multiclass classification task and the data for the binary classification tasks remain unchanged throughout the iterative process. The same strategy is recently used by (Schick and Schütze, 2020) to create small language models that have similar performance to BERT and (Xie et al., 2020) to achieve state-of-the-art performance on Imagenet in computer vision.

Adding up the techniques described above, we denote this model as SS MT D/TAPT ALBERT.

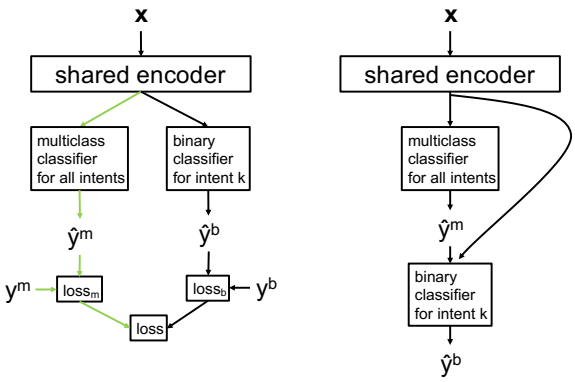

Figure 2: Training (left) and inferencing (right) for the multi-task learning strategy, where $k \in [0...n-1]$, and n is the number of intent classes. In training time, the green path is only executed when **x** is a positive sample.

4 Experiments and Results

4.1 Data and Experimental Setup

For confidentiality reasons, we can only share directional numbers about the training data. In this study, we consider 9 customer intent classes. The curated data is unbalanced among classes, ranging from a few thousand to tens of thousands of records per class. The class with the most samples is roughly 40 times as much as the class with the least samples. For each class, the ratio of positive-to-negative cases in the curated data is about 4. The un-curated data is roughly 20 times of the curated data. We use both the curated and un-curated data for DAPT and only curated data for TAPT. In the semi-supervision process, for each class, we select high-confidence samples from the un-curated data in each iteration to be roughly two to three times of the volume of the labeled samples in the curated data. Table 1 shows a few sample training data with dummy features and intents. The last column

Table 1: Sample training data and how different training strategies incorporate them.

Curation	Composition	Features	Targets		Training data for				
		Messages	Intents	CSR responses	Multiclass task	Binary tasks	DAPT	TAPT	SS
Curated	Positives	Could you help me?	General inquiry	Yes	Yes	Yes (+)	Yes	Yes	Yes
		How to setup account?	Account issue	Yes	Yes	Yes (+)	Yes	Yes	Yes
	Negatives	How much is this?	Account issue	No	No	Yes (-)	Yes	Yes	Yes
		Can you fix this issue?	General inquiry	No	No	Yes (-)	Yes	Yes	Yes
Un-curated	Unlabeled	What's this?	General inquiry	N/A	No	No	Yes	No	Yes
		Please help.	N/A	N/A	No	No	Yes	No	Yes

shows how different strategies incorporate them in training.

After being processed with the ALBERT tokenizer, the total data amounts to about 800 million tokens with an average of about 80 per sample. We performed all experiments on Sagemaker on AWS. We used 2 ml.p3.16xlarge instances with distributed data parallelism for DAPT and TAPT, 1 ml.p3.8xlarge instance for finetuning language models, and 1 ml.p3.8xlarge instance for batch inferencing testing data.

We hold out a portion of the data as development data to tune hyperparameters. We follow the suggestions from (Gururangan et al., 2020; Liu et al., 2019) for DAPT and TAPT and (Devlin et al., 2019) for finetuning. For the end-to-end multi-task learning process, we kept a unit weight for each task and did not explore different weight combinations. More research about tuning task weights in multi-task learning can be found in (Cipolla et al., 2018).

4.2 Evaluation

4.2.1 Pretrained models

In this section, we evaluate the performance of the pretrained language models, the out-of-the-box ALBERT and the D/TAPT ALBERT. We note that the pretrained language models are evaluated before any finetuning happens.

To visually demonstrate how the adaptive pretraining improves the clustering performance of the classification vector, we sample a couple thousand cases per class and apply t-SNE (Van Der Maaten and Hinton, 2008) to the reduced classification vector for each case. We reduce the dimension of the classification vectors from 768 to 50 with PCA to keep the computational cost of t-SNE in check. In Figure 3, we can see how clustering improves from the vanilla ALBERT on the left to D/TAPT ALBERT on the right.

To more quantitatively assess the performance of the off-the-shelf pretrained ALBERT and the D/TAPT ALBERT, we sample a couple thousand

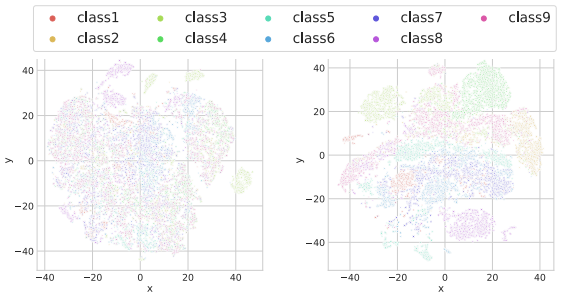

Figure 3: t-SNE plots using the dimension-reduced classification vectors from the off-the-shelf pretrained ALBERT (left) and the D/TAPT ALBERT (right).

Table 2: Average kNN prediction accuracy using the classification vectors from the pretrained models

ALBERT	D/TAPT ALBERT
0%	+33%

cases per class and use k-nearest-neighbor classifiers (kNN) to predict each sample's class based on its k neighbors. We use the Euclidean distance between the classification vector for each case as the similarity metric for kNN. We compute the average accuracy by varying k from 3 to 99 in interval of 2 and report it in Table 2. As a result, D/TAPT lifts the accuracy by more than 30 points compared to the vanilla ALBERT. Similar performance lift is also observed in (Reimers and Gurevych, 2019). This illustrates that D/TAPT can improve the clustering performance of the classification vector when the clustering rules are closely related to the domain corpus. The absolute accuracy values are not reported here for confidentiality reasons.

4.2.2 Finetuned models

In practice, for each class, we expect to route more positive cases and less negative cases to our CSRs with machine learning models. That means we expect our models to better differentiate positives from negatives for each class. Area Under the Curve - Receiver Operating Characteristics (AUC

Table 3: The average and sample-weighted average AUC ROC for different experiment settings

Model	avg AUC ROC	wavg AUC ROC
ALBERT	+0%	+0%
+ MT	+17.8%	+14.3%
+ MT DAPT	+18.4%	+ 15.8%
+ MT D/TAPT	+19.0%	+16.1%
+ SS MT D/TAPT	+19.9%	+17.0%

ROC) is a natural metric for such binary classification problem. We note that the commonly-used *accuracy* metric is not appropriate in this context since the negatives do not have ground truth labels in our data. The evaluation data is from recent few weeks. For confidentiality reasons, we hide the axis for AUC ROC and make the values relative to the baseline finetuned ALBERT model for each class.

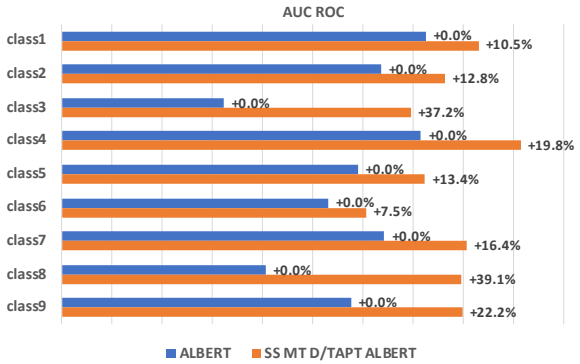

Figure 4: The AUC ROC of each class for finetuned ALBERT and SS MT D/TAPT ALBERT

In Figure 4, for each class, we can observe consistent improvement of SS MT D/TAPT ALBERT over finetuned ALBERT in terms of AUC ROC. Overall, the SS MT D/TAPT ALBERT model brings 19.9 points increase in average AUC ROC and 17 points increase in sample-weighted average AUC ROC, compared to the finetuned ALBERT model.

Furthermore, it is interesting to see how each strategy in the SS MT D/TAPT ALBERT model contributes to the performance improvement. In Table 3, we show the average and sample-weighted average AUC ROC improvement by incrementally adding one strategy at a time. We can see that the MT strategy boosts the average AUC ROC by 17.8 points and the sample-weighted average AUC ROC by 14.3 points, compared to the finetuned ALBERT. This demonstrates the effectiveness of including negative signals with MT strategy. On top of MT, we apply DAPT, D/TAPT and SS incrementally. Each strategy pushes up the average and sample-weighted average AUC ROC by roughly 1 point.

5 Discussion

Apart from processing the dismissed recommendations with this multi-task setting, there is another heuristic approach that is commonly adopted under this circumstance. We can group all the dismissed recommendations into an extra bucket *Others* (Fotso et al., 2018). The advantage of this approach is that we can pose the problem as a straightforward multiclass classification. The disadvantage is that the dismissed recommendations can either be mis-classified and belong to other existing classes, or belong to unknown classes that might be included in the future taxonomy. In the former scenario, the dismissed recommendations create noise for their true class and the *Others* class; In the latter scenario, the dismissed recommendations can seemingly improve performance for current taxonomy, while they can pollute the future training when the unknown classes are launched in the updated taxonomy. In both scenarios, grouping the dismissed recommendations into *Others* can negatively impact the training.

In terms of computational cost, both adaptive pretraining and semi-supervision consume a considerable amount of power, since the former is typically trained on the MLM task through a large corpus and the latter is a iterative finetuning and inferencing process where the data for inferencing are often in large volume. In the meantime, the MT strategy is a cost-effective way to improve model performance by considering negative samples. By examining Table 3, compared to the baseline finetuned ALBERT, we can see the MT strategy increases the average AUC ROC by 17.8 points while D/TAPT and SS add 2.1 points on top of that. The additional cost for the MT strategy, compared to the typical multiclass classification strategy, is simply a binary classifier for each class. It is negligible in both training and inferencing.

For the sake of easy implementation of the end-to-end multi-task training, we only feed the training data related to one task in each batch. In this way, we can keep the loss function for each task separate. It is possible that including data for various tasks in each batch can bring benefits to training. This assumption can be explored in future studies.

This study is only concerned with corpus in En-

glish. Similar modeling strategies can be followed for other high-resource languages which we have ample training data. However, as in the customer service departments of most global organizations, it is common to receive customer contacts in various low-resource languages, in which case the training data is scarce. Recent advances in cross-lingual language models, such as mBERT (Devlin et al., 2019), XLM (Conneau and Lample, 2019), Unicoder (Huang et al., 2019) and FILTER (Fang et al., 2020), can shed light on this situation and we plan to investigate it in the future.

In the area of customer support, both (Molino et al., 2018) and (Fotso et al., 2018) propose neural networks that combine unstructured text features from customers' messages and structured features describing customers' interaction with the platforms. They empirically demonstrated benefits of including the latter feature group. The next step for our study is to evaluate the influence of the customer-website interaction features, when combining with advanced language models.

For the model candidates with multi-task strategy in this manuscript, we train all tasks jointly with an end-to-end multi-task deep learning approach, as described in the left plot of Figure 2. We want to point out the isolating effect of the end-to-end training approach. In one experiment, we trained the tasks independently, i.e. we first trained the multiclass classification task with the off-the-shelf ALBERT, and then, for the binary tasks, we trained n logistic regression binary classifiers with the classification vector from the multiclass classification task. We still achieved 12.2 and 8.2 points above the baseline in terms of average AUC ROC and sample-weighted average AUC ROC. On one hand, this shows that even training simpler models independently can still bring performance lifts, thus emphasizing the powerful signal brought by the negative cases; On the other hand, if compared to the ALBERT + MT model in Table 3, it also shows the benefits of end-to-end training.

As in most machine learning applications, the actual model performance is determined by the choice of the operational point for each intent class and the operational point is determined from the precision-recall (PR) curve. For the sake of brevity, we ignore the PR plots because, for each class, the PR curve of the baseline ALBERT model is well under the envelop of the PR curve of the SS MT D/TAPT ALBERT model. This is expected due to the large boost presented in Figure 4. We note that the AUC ROC can be a decent indication of AUC PR when the data is not so skewed (Davis and Goadrich, 2006). Therefore, the SS MT D/TAPT ALBERT indeed outperforms the baseline for every choice of operational point.

6 Conclusion

In this manuscript, we demonstrated and discussed the model performance improvement brought by multi-task learning, adaptive pretraining for ALBERT, and semi-supervised learning in the application of customer support on an e-commerce website. We observe $\sim$20 points performance increase in average AUC ROC when comparing the final model to the baseline multiclass classification model. This paradigm can be particularly helpful when there is a feedback system collecting confirmation from labelers. Future studies can extend this paradigm to more complex situations, such as when the intent taxonomy is deeply hierarchical or considering more feedback information than simple "yes" or "no".

Acknowledgments

The authors wish to thank Harsha Aduri and Jieyi Jiang for providing support in data preparation, Jasmine Qi and Ilknur Egilmez for providing comments for the manuscript. We also thank the anonymous reviewers for their valuable suggestions.

References

Roberto Cipolla, Yarin Gal, and Alex Kendall. 2018. Multi-task Learning Using Uncertainty to Weigh Losses for Scene Geometry and Semantics. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7482–7491, Salt Lake City, UT, USA.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators. *arXiv preprint*.

Alexis Conneau and Guillaume Lample. 2019. Cross-lingual Language Model Pretraining. In *Advances in Neural Information Processing Systems*, volume 32, pages 7059–7069. Curran Associates, Inc.

Jesse Davis and Mark Goadrich. 2006. The relationship between precision-recall and ROC curves. In *ACM International Conference Proceeding Series*, volume 148, pages 233–240.

Alexandra DeLucia and Elisabeth Moore. 2020. Analyzing HPC Support Tickets: Experience and Recommendations. *arXiv preprint*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. 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, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Charles Elkan and Keith Noto. 2008. Learning Classifiers from Only Positive and Unlabeled Data. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '08, pages 213–220, New York, NY, USA. Association for Computing Machinery.

Yuwei Fang, Shuohang Wang, Zhe Gan, Siqi Sun, and Jingjing Liu. 2020. FILTER: An Enhanced Fusion Method for Cross-lingual Language Understanding. *arXiv preprint*.

Geli Fei and Bing Liu. 2015. Social Media Text Classification under Negative Covariate Shift. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2347–2356, Lisbon, Portugal. Association for Computational Linguistics.

Stephane Fotso, Philip Spanoudes, Benjamin C. Ponedel, Brian Reynoso, and Janet Ko. 2018. Attention fusion networks: Combining behavior and E-mail content to improve customer support. *arXiv preprint*.

Zhenxin Fu, Shaobo Cui, Mingyue Shang, Feng Ji, Dongyan Zhao, Haiqing Chen, and Rui Yan. 2020. Context-to-Session Matching: Utilizing Whole Session for Response Selection in Information-Seeking Dialogue Systems. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, volume 20, pages 1605–1613.

Narendra Gupta, Mazin Gilbert, and Giuseppe Di Fabbrizio. 2010. Emotion Detection in Email Customer Care. In *Proceedings of the NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text*, pages 10–16, Los Angeles, CA. Association for Computational Linguistics.

Narendra Gupta, Gokhan Tur, Dilek Hakkani-Tur, Srinivas Bangalore, Giuseppe Riccardi, and Mazin Gilbert. 2006. The AT&T spoken language understanding system. *IEEE Transactions on Audio, Speech, and Language Processing*, 14(1):213–222.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A Smith. 2020. Don't Stop Pretraining: Adapt Language Models to Domains and Tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360, Online. Association for Computational Linguistics.

Yu-Guan Hsieh, Gang Niu, and Masashi Sugiyama. 2019. Classification from Positive, Unlabeled and Biased Negative Data. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2820–2829, Long Beach, CA, USA. PMLR.

Haoyang Huang, Yaobo Liang, Nan Duan, Ming Gong, Linjun Shou, Daxin Jiang, and Ming Zhou. 2019. Unicoder: A Universal Language Encoder by Pretraining with Multiple Cross-lingual Tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2485–2494, Hong Kong, China. Association for Computational Linguistics.

Schubert Cheung Hui and G. Jha. 2000. Data mining for customer service support. *Information and Management*, 38(1):1–13.

Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. 2017. RACE: Large-scale ReAding comprehension dataset from examinations. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 785–794, Copenhagen, Denmark. Association for Computational Linguistics.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint*.

Xiao-Li Li, Bing Liu, and See-Kiong Ng. 2010. Negative Training Data Can be Harmful to Text Classification. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, 11, pages 218–228, Cambridge, MA, USA. Association for Computational Linguistics.

Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. 2019. Multi-Task Deep Neural Networks for Natural Language Understanding. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4487–4496, Florence, Italy. Association for Computational Linguistics.

Senthil Mani, Neelamadhav Gantayat, Rahul Aralikatte, Monika Gupta, Sampath Dechu, Anush Sankaran, Shreya Khare, Barry Mitchell, Hemamalini Subramanian, and Hema Venkatarangan. 2018. Hi, How Can I Help You?: Automating Enterprise IT Support Help Desks. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications*

of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), pages 7688–7695, New Orleans, Louisiana, USA, February 2-7, 2018. AAAI Press.

Piero Molino, Huaixiu Zheng, and Yi Chia Wang. 2018. COTA: Improving the speed and accuracy of customer support through ranking and deep networks. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 4:586–595.

Michael Powell, Jamison A Rotz, and Kevin D O'Malley. 2020. How Machine Learning Is Improving U.S. Navy Customer Support. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 13188–13195.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuad: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Alan Ramponi and Barbara Plank. 2020. Neural Unsupervised Domain Adaptation in NLP—A Survey. *arXiv preprint*.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *arXiv preprint*.

Ruhi Sarikaya, Geoffrey E Hinton, and Bhuvana Ramabhadran. 2011. Deep belief nets for natural language call-routing. In *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5680–5683, Prague, Czech Republic. IEEE.

Timo Schick and Hinrich Schütze. 2020. It's Not Just Size That Matters: Small Language Models Are Also Few-Shot Learners. *arXiv preprint*.

Laurens Van Der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2625.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2019a. SuperGLUE: A Stickier Benchmark for General-Purpose Language Understanding Systems. In *Advances in Neural Information Processing Systems*, volume 32, pages 3266–3280. Curran Associates, Inc.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Wei Wang, Bin Bi, Ming Yan, Chen Wu, Zuyi Bao, Jiangnan Xia, Liwei Peng, and Luo Si. 2019b. StructBERT: Incorporating Language Structures into Pre-training for Deep Language Understanding. *arXiv preprint*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Transformers: State-of-the-art natural language processing. *arXiv preprint*.

Patrick Xia, Shijie Wu, and Benjamin Van Durme. 2020. Which *BERT? A Survey Organizing Contextualized Encoders. *arXiv preprint*.

Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V. Le. 2020. Self-Training With Noisy Student Improves ImageNet Classification. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10684–10695, Seattle, WA, USA.

Puyang Xu and Ruhi Sarikaya. 2013. Convolutional neural network based triangular CRF for joint intent detection and slot filling. In *2013 IEEE Workshop on Automatic Speech Recognition and Understanding*, pages 78–83, Olomouc, Czech Republic. IEEE.

Yixing Xu, Chang Xu, Chao Xu, and Dacheng Tao. 2017. Multi-Positive and Unlabeled Learning. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, (IJCAI-17)*, pages 3182–3188, Melbourne, Australia.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Russ R Salakhutdinov, and Quoc V Le. 2019. XLNet: Generalized Autoregressive Pretraining for Language Understanding. In *Advances in Neural Information Processing Systems*, volume 32, pages 5753–5763, Vancouver, Canada. Curran Associates, Inc.