SAS-BERT: BERT FOR SALES AND SUPPORT CONVERSATION CLASSIFICATION USING A NOVEL MULTI-OBJECTIVE PRE-TRAINING FRAMEWORK

Aanchal Varma and Chetan Bhat

Fresh works, India

ABSTRACT

Recent emergence of large language models (LLMs), particularly GPT variants has created a lot of buzz due to their state-of-the-art performance results. However, for highly domain-specific datasets such as sales and support conversations, most LLMs do not exhibit high performance out-of-the-box. Thus, fine-tuning is neededwhich many budget-constrained businesses cannot afford. Also, these models have very slow inference times making them unsuitable for many real-time applications. Lack of interpretability and access to probabilistic inferences is another problem. For such reasons, BERT-based models are preferred.

In this paper, we present SAS-BERT, a BERT-based architecture for sales and support conversations. Through novel pre-training enhancements and GPT-3.5 led data augmentation, we demonstrate improvement in BERT performance for highly domain-specific datasets which iscomparable withfine-tuned LLMs.

Our architecture has 98.5% fewer parameters compared to the largest LLM considered, trains under 72 hours, and can be hosted on a single large CPU for inference.

KEYWORDS

BERT, LLM, Text Classification, Domain pre-training, NLP applications

1. INTRODUCTION

Recent innovations in large language models (LLMs) have gained significant attention and popularity in industry and academia alike. The latest round of LLMs such as GPT-4[1], and GPT-3[2], among others, consist of hundreds of billions of trained parameters and have achieved impressive performance in many NLP tasks using in-context learning and prompt engineering—a few-shot learning paradigm first introduced by [2].This learning paradigm allows these LLMs to use their natural language generation capabilities to solve any task, by completing a piece of text or prompt. Along with open-source contributions occurring at an overwhelming pace, examples including models such as Falcon-40B [3], MPT-7B, and Llama 65B [4], the LLM and NLP ecosystem continues to transform at a rapid pace.

These LLMs, however, are not without limitations. First, most of these models require large GPUs to load and host, which is already in the realms of infeasibility for most budget-constrained small and medium-sized businesses. Similar cost constraints arise with paid APIs offered by

David C. Wyld et al. (Eds): AIBD, MLSC, ACSTY, NATP, CCCIoT, SVC, SOFE, ITCSS -2024
pp. 239-252, 2024. CS & IT - CSCP 2024 DOI: 10.5121/csit.2024.140419

other companies that abstract away the serving infrastructures. Second, there has been growing evidence that these LLMs are not very impressive on niche, domain-specific tasks, as shown in [5], where fine-tuned BERT models even outperform few-shot inference using GPT4. This necessitates extensive fine-tuning of these models for domain-specific tasks, where costs can quickly add up to hundreds of thousands of dollars. Techniques such as adaptor tuning, and LoRA have been developed to enable faster parameter efficient fine-tuning of large models as shown in [6]. However, these techniques still suffer from diminishing results and are not enough to offset the cost constraints. Third, most LLMs have terribly slow inference times, making them unsuitable for real-time production systems that have stringent latency requirements. Finally, most businesses especially in the SaaS industry rely on accurate sales deal outcome predictions. To plan and strategize well, these businesses need prediction probability distributions, supplemented with good explainability for these predictions. BERT family of models seem better suited for these requirements and have thoroughly been tested over the years. Given these limitations with LLMs as well as strategic business requirements, smaller transformer architectures using attention [7], like BERT [8] are still a promising alternative, which this paper goes on to explore deeply. Additionally, recent work on data augmentation via generative models by [9] showed that using GPT to create a small but clean training dataset to augment the training process can help bolster model performance. We take inspiration from this and utilize this idea in our work here.

We present a two-stage BERT training regimen in this paper. In the first stage, we further pre-train BERT using a multi-objective training architecture, on a rich collection of in-domain corpora consisting of email and support conversations. We unlock significant improvements by tapping into the conversational dynamics of these conversations using this pre-training setup. The first objective in this multi-objective framework is an enhancement to the masked language modelling (MLM) technique, where we make two enhancements - a. Instead of masking tokens randomly, we mask important tokens identified using a separate Naive-Bayes model. b. In addition to masking the chosen token, we mask a span of tokens around it. These enhancements were inspired from separate work presented in [10] and [11]. The second objective is a variant of next sentence prediction (NSP), where instead of next sentence, we train the model on conversation pairs, so that it learns to predict if one email (or support response) follows the preceding email (or support question) or not. In the second stage of the training regimen, we fine-tune BERT for conversation classification. We augment the training data used in this task with a small but clean dataset of email conversations generated using GPT-3.5. We utilize our internal CRM data collected over several years for the training and fine-tuning steps. The details of the data used is covered in Section 3.1.

Through this two-stage training process, we demonstrate that our overall architecture demonstrates comparable performance as some of the fine-tuned LLMs, trains in fraction of the time, and achieves these noteworthy results at a fraction of the cost. We plan to release our model, SAS-BERT, so that the larger community can reap better predictive performances on their domain-specific sales and support conversations. Additionally, we also plan to release our model & training code so that our proposed enhancements may be applied to other domains, datasets, and industries.

The remainder of this paper is structured as follows: In Section 2, we provide an overview of the related work in the field. Section 3 describes our model architecture and presents each aspect of our training and fine-tuning framework in detail. In Section 4, we discuss the results from our experiments in terms of performance and quality of embeddings generated from our custom model. In section 5, we provide a detailed ablation study demonstrating the incremental lift in performance from each of the enhancements considered. Finally, we conclude our paper and outline future possibilities for this framework.

2. RELATED WORK

Domain pre-training of language models and transformer architectures have been quite popular ever since the launch of pre-trained models such as BERT [8]. RoBERTa [12] explores the ideas of dynamic masking and experiments with BERT pre-training in terms of hyperparameters and training data size. XLM[13] explores pretraining for multilingual dataset that can cater to different languages across the globe. XLNet[14] uses autoregressive loss and transformers-XL architecture with increased data size over BERT to show results on multiple benchmarks. XLNet also masks spans of tokens, but the masking happens in an autoregressive manner.Work in SCIBERT [15] and PubMedBERT [16] and other similar work in the past attempt to train transformer architectures from scratch. They argue that for domains with abundant unlabeled text such as biomedicine, it is unclear that domain-specific pre-training can benefit by transfer from general domains. Most of the general domain text is substantively different from biomedical text, raising the prospect of negative transfer that can hinder the target performance. We follow a similar approach as in BioBERT [17] where we further train the BERT model that was originally trained on general domain data. We believe that word relationships learnt in the general domain are still very much applicable in the sales and support conversations domain, thus our pre-training approach aims to retain this learning, while developing stronger representations for in-domain keywords. In other work, task-specific corpora, like NER datasets, have been used to pre-train BERT as shown in NER BERT [18].

Domain pre-training is being explored across different domains such as legal Legal-BERT[19], finance Fin-BERT [20], aviation Aviation-BERT [21] and many more areas. While in the medical domain, pretraining has also been explored on different languages such as Danish MeDa-BERT[22], German MedBERT [23] and other languages as well.

There has also been work done to either extend the pre-training architectures with additional layers of learning or introduce novel loss paradigms and masking techniques during pre-training. For example, ExBERT [24] extends BERT by augmenting its embeddings for the original vocabulary with new embeddings for the domain-specific vocabulary via a learned small "extension" module. In SpanBERT [11], BERT is pre-trained by (1) masking contiguous random spans, rather than random tokens, and (2) training the span boundary representations to predict the entire content of the masked span, without relying on the individual token representations within it. In NB-MLM [10], the authors demonstrate that MLM objective leads to inefficiency because it mostly learns to predict the most frequent words. They propose a technique for more efficient adaptation that focuses on predicting words with large weights of the Naive Bayes classifier. We take inspiration from [10] and [11] to custom design our MLM objective.

In MacBERT [25], pretraining has been explored using whole word masking on Chinese dataset. Models such as DistilBERT[26], ALBERT [27] propose techniques to reduce parameters while training for faster training and lower memory consumptions. Models like StructBERT[28] tend to include language structures while pretraining at both word and sentence level. ConvBERT [29] proposes a span based dynamic convolution technique to directly model local dependencies.

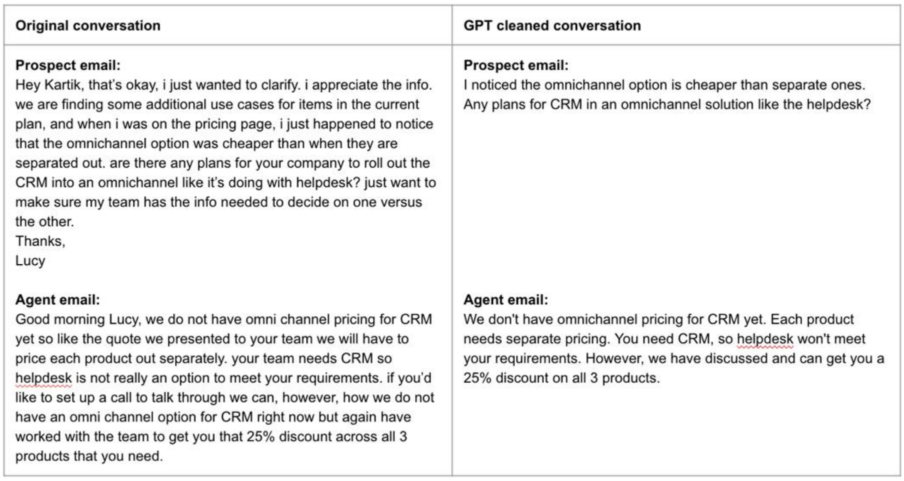
Additionally, there has been work like in TOD-BERT [30], and in CS-BERT [31], that use conversational data to pre-train BERT for conversational tasks. We leverage the conversational data available to us which is unique to sales and support conversations and design a next email/support response prediction objective around this data. The strength of our pre-training architecture comes from synergistic and compounding effects we found by having multiple conceptual improvements in the same set-up.

3. EXPERIMENTAL SETUP

This section covers details of the datasets used in the training experiments, LLMs and off-the-shelf BERT models considered, and specifics on the multi-objective pre-training as well as fine-tuning set up of our final architecture.

3.1. Datasets

Sales conversations play a crucial role in CRM to identify the likelihood of a win. These conversations play a crucial role to capture variations and many flavours in sales deals negotiation styles, as well as issue addressals in the form of customer support. We leverage this dataset of sales conversations happening via emails and support ticket conversations, to identify meaning and infer more on the deal closure reasons. We use different datasets belonging to these email conversations for various aspects of training our custom architecture. The details of these datasets are included in Table 1. Important point to note is that all the datasets are mutually exclusive, and do not contain any overlaps of emails and support conversations between datasets. We do this to avoid any type of leakage of information during the different learning protocols. As mentioned earlier, we have extracted emails and support ticket conversations from our CRM database spanning several years and thousands of businesses. These datasets are at a sales deal level, where a deal in the CRM context refers to an entity created by the sales agent in the CRM for a prospective customer, indicating an active ongoing sales process with that prospect. The datasets used are conversations mapped to deals, where these conversations can be either email conversations around selling and negotiations, or support conversations addressing product issues, between prospects and agents. A lot of these conversations revolve around discussions on explaining the product, scheduling demo, discussing on pricing, identifying customer requirements while trying to close a sales deal. These could also be around support conversations navigating issues with products, helping with additional feature requests, and guiding through right usage of the products. The deals eventually go on to win or lose (1 or 0). This corresponds to the target label, and this information is used in the fine-tuning step where the task is to classify these conversations with the objective of minimizing the classification loss.This kind of training on email conversations help us evaluate the closure probabilities for emails in ongoing deals, which further help in understanding sales deals which could be on the verge of losing, or even help us in identifying high priority deals where customers are readily interested.



Original conversation	GPT cleaned conversation
Prospect email: Hey Kartik, that's okay, i just wanted to clarify. i appreciate the info. we are finding some additional use cases for items in the current plan, and when i was on the pricing page, i just happened to notice that the omnichannel option was cheaper than when they are separated out. are there any plans for your company to roll out the CRM into an omnichannel like it's doing with helpdesk? just want to make sure my team has the info needed to decide on one versus the other. Thanks, Lucy	**Prospect email:** I noticed the omnichannel option is cheaper than separate ones. Any plans for CRM in an omnichannel solution like the helpdesk?
Agent email: Good morning Lucy, we do not have omni channel pricing for CRM yet so like the quote we presented to your team we will have to price each product out separately. your team needs CRM so helpdesk is not really an option to meet your requirements. if you'd like to set up a call to talk through we can, however, how we do not have an omni channel option for CRM right now but again have worked with the team to get you that 25% discount across all 3 products that you need.	**Agent email:** We don't have omnichannel pricing for CRM yet. Each product needs separate pricing. You need CRM, so helpdesk won't meet your requirements. However, we have discussed and can get you a 25% discount on all 3 products.

Figure 1. Example of actual email exchange between a prospect and a sales agent, and the GPT-cleaned exchange

This target label is also included in the test set which is used to measure the classification F1 score for all the models considered in our experiments.

As described earlier, one of the enhancements in our training architecture is the inclusion of a small and clean dataset during the fine-tuning process. We leverage GPT-3.5 for this purpose. To keep the costs in check, we clean only 20k emails. Although small, this inclusion gives us surprisingly good increments to the model's predictive performance as can be seen in the results section. Figure 1 shows two examples of clean conversation generated from GPT-3.5. To do this, we picked existing email conversations that fit within GPT-3.5 prompt token limits, and prompted it to modify the conversation, without losing context and information. Examples of some modifications were to remove irrelevant sections of the conversations, removing disclaimers and signatures, rewording emails, replacing with synonyms, correcting for spelling and grammar etc. For naming consistency, in the rest of the sections, we simply refer to our datasets as "email conversations" or "emails" for brevity.

Table 1. Description of the datasets, presence of target label and the size of dataset used for various parts / aspects of the custom training architecture.

Datasets	**Target label?**	**Dataset size**
Pre-training dataset	No	8M
Dataset to identify Naive Bayes tokens for masking	Yes	500k
Fine-tuning dataset	Yes	400k
GPT-3.5 cleaned dataset for fine-tuning augmentation	No	20k
Test dataset	Yes	15k

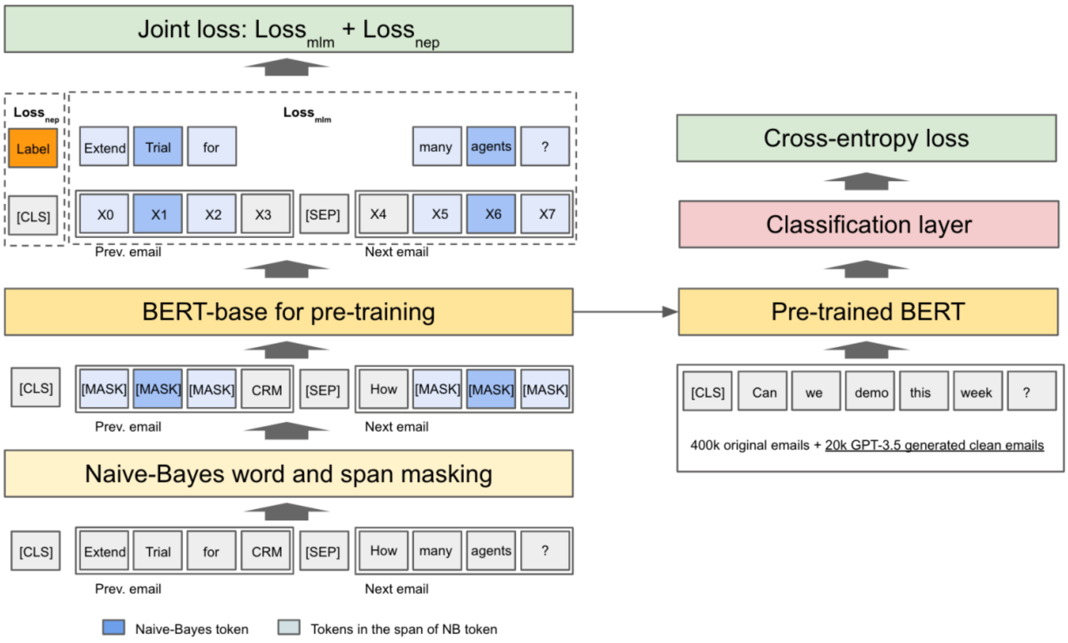
3.2. LLM Models

We evaluated several LLMs as well as BERT-based off-the-shelf models to set up an expansive baseline for this paper. Here we evaluated models such as Falcon, Llama, OPT, Bloom and GPT models which are trained to generate next best word based on the context. For some of the LLMs, we have followed two approaches - a. Evaluating them off-the-shelf b. Fine-tuning on a larger dataset with parameter optimization techniques like LoRA [32]. We stack up these different results and compare our custom architecture's performance against these. Among many LLM models available for training today, we have picked those that have had notable mentions in literature and did not pose memory issues during the training process.

Since LLMs do not have a specific classification layer and a classification loss that can emit a probability of deal closure, we prompt the LLM with a simple prompt as follows - "You are a sales deal closure prediction bot. You will be given a sales email or a support ticket body of text corresponding to either a sales agent or a prospective customer. Given this piece of context, you need to predict, as accurately as possible, if the deal associated with that prospective customer will win or lose. If you predict the deal will win, return yes. If you predict the deal will lose, return no". With careful crafting of prompts, we ensure to get 'yes'/'no' output in most cases and ignore a few misses by LLMs. We then take LLMs responses "yes" and "no" to be the prediction values 1 and 0. Since our prediction values are restricted to just these two values and do not have a probability in the range 0-1, we did not choose AUC as the evaluation metric. We chose F1 instead.Likewise, for the BERT based off-the-shelf models, we picked those ones that most feature in literature, also making sure there is variety in their underlying training paradigms and training datasets used.

3.3. Our Multi-Objective Training Framework

We propose a novel pre-training framework which focuses on masking spans of relevant tokens, along with next email prediction task. Once the models are pre-trained on our sales conversations and have better understanding of the context, they are finetuned for downstream classification tasks.



(a) Pre-training (b) Fine-tuning

Figure 2. This figure represents our two-stage BERT architecture. The architecture to the left (a) depicts the pre-training setup with Naive-Bayes token and span masking for the MLM task, as well as the next email prediction task, with both losses jointly optimized. The architecture to the right (b) shows the fine-tuning setup, which uses pre-trained BERT for conversation classification. This step is augmented with clean emails data generated using GPT-3.5

3.3.1. BERT Pretraining

We take an off-the-shelf BERT model (BERT base uncased) and set up pre-training and fine-tuning as shown in figure 2. The rest of this sub-section goes on to detail this multi-objective setup.

Naive Bayes MLM:

Our first objective is a modification to BERT's masked language modelling (MLM) technique. The MLM objective used in BERT pre-training is a cross-entropy loss for predicting masked tokens. The masking technique uniformly selects 15% of the input tokens to be masked, and 80% of these masked tokens are replaced with [MASK] token, while 10% are left unchanged, and 10% are replaced randomly. Past work has shown that this technique can be further improved by masking only those tokens that are highly relevant to the domain, thereby learning better representations for them. In our case, keywords such as 'helpdesk', 'omnichannel', 'trial', 'support', 'demo', 'purchase', 'subscription' etc. are relevant to sales and support conversations domain especially in SaaS settings and can benefit from stronger learned representations. As detailed in the datasets section, to obtain such highly relevant keywords for masking, we run a Naive Bayes classifier on a separate corpus of email conversations with deal won/lost being the target label.

A Naive Bayes classifier allows assigning conditional probabilities to tokens which demonstrate their importance towards predicting the health of the deal, which simpler frequency-based word cloud models don't provide. Also, these scores allow for using score thresholds to shortlist a desired number of tokens for masking.

Using the token-level conditional probabilities resulting from this classifier, we arrive at the importance score of each token as below –

$$fi(w) = |logP(w|1) - \log P(w|0)| \tag{1}$$

Where 0 and 1 are our deal closure labels, and w is each token for which we compute the importance score. We pick candidate tokens for masking that have a score above a certain threshold score.

This way, we pick the words which hold the most relevance for each class and masking them enables our models to learn better representations especially for these relevant keywords.

Figure 3 shows the top Naive Bayes identified token for masking. The values refer to the importance scores assigned to them by the model.

A second enhancement to this objective is masking a span of tokens around the Naive Bayes identified tokens. It has been observed that masking contiguous tokens of text allows better context learning rather than individual tokens especially in cases where bigrams and trigrams themselves are more relevant [11]. Following the paper, at each iteration, we first sample a span length (number of words) from a geometric distribution ~Geo(p), which is skewed towards shorter spans. We fix p as 0.2 and clip the span length to a maximum of 10, based on analysis done in [11].

Mathematically, optimizing this span based NBMLM loss can be represented as below –

$$L_{MLM} = -\sum_{i=1}^{N} logP(xi|X) \tag{2}$$

where x_i is a masked token in the span length, and we predict the probability over the set of tokens X = $\{x_1, x_2, \ldots x_N\}$, with N being the total vocab size.

Our token masking strategy is largely like that used in BERT's MLM task. To pick 15% tokens to be masked, our masking module first identifies all Naive-Bayes tokens and the span of tokens around it. If this constitutes 15% of all tokens or more, up to 15% of the tokens are masked. If it is less than 15%, then the remaining tokens are picked randomly from the dataset. Rest of the masking strategy is the same as that implemented in BERT [8].

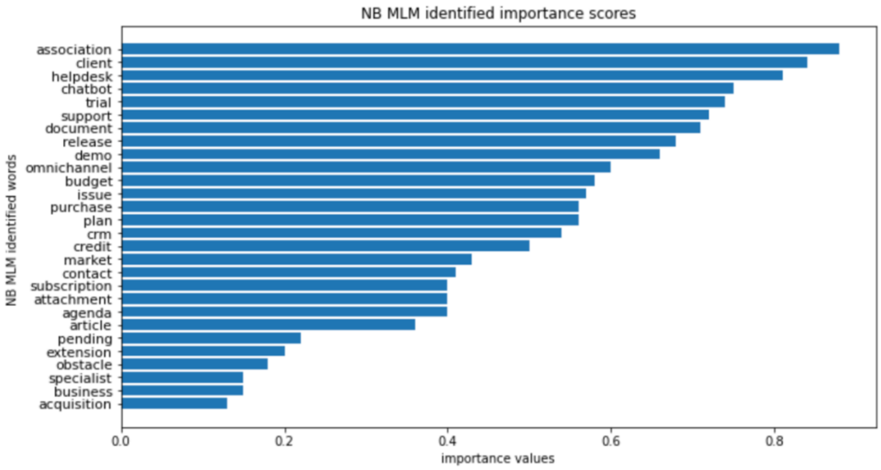


Figure 3: Naive Bayes identified tokens and their conditional probabilities which show how relevant these words are with reference to our corpus.

Next Email Prediction (NEP):

The second objective involves a modified version of next sentence prediction (NSP), where we task the model to predict the next email. That is, we supply the network with email pairs, and task it to predict if one email follows the other email or not.

We take our existing dataset of email conversations and create email pairs from them. To create positive pairs, we pick agent and customer email pairs that occurred one after the other, with some basic pre-processing to remove irrelevant and uninformative emails. For negative pairs, we randomly pick conversations from different deals, so we can generate sufficient negative examples.

The NEP loss is a cross-entropy loss calculated against the positive and negative labels associated with the email pairs.

$$L_{NEP} = \left(-y_j \log\left(p_j\right)\right) + \left(\left(1 - y_j\right) \log\left(1 - p_j\right)\right) \quad (3)$$

Where pj is the predicted probability of the second email following the first in pair j, and yj is the ground truth.

The network is trained to minimize the sum of the two losses i.e.,$L_{MLM}$and L_{NEP}.

3.3.2. BERT Finetuning

After pre-training the BERT, we proceed to fine-tuning this model. This step is the standard fine-tuning process by adding a classification layer to the BERT model. As explained earlier, the GPT-3.5 cleaned dataset of ~20k examples was appended with the rest of the fine-tuning set of ~400k emails. This fine-tuned model was evaluated on a test set containing only original emails, that were not seen either during the pre-training or the fine-tuning step.

4. RESULTS

The pre-training task is carried out for ~8M sales email conversations which are fed to the custom architecture in the form of pairs, along with a label to identify if the pair is a valid pair or not. This set up is necessary for the next email prediction task. Within this pairing, NB tokens are identified for masking, along with masking a span of tokens around them the identified. We obtained ~20k relevant tokens for masking from the Naive Bayes classifier, which was run on a separate email conversations dataset containing 500k emails. This architecture is trained for about 4 epochs, on g5.24xlarge GPU, with a batch size of 16, using gradient accumulation size of 2, for 3 days. This pre-trained model is checkpointed and then used for the fine-tuning task in the next step.

Fine-tuning was conducted on an email conversation corpus of ~400k emails. We cleaned about ~20k emails with GPT-3.5 and included it in the fine-tuning set. The architecture was trained for 5 epochs with a batch size of 128 for about 5 hours. The fine-tuning loss is binary cross entropy loss, predicting deal closure probabilities. The fine-tuned model was tested on a test set of ~15k sales email conversations. As explained earlier, the GPT-3.5 cleaned dataset was added only to the fine-tuning training set. Test set comprises original emails only, and no GPT-3.5 cleaned emails were added to it. We chose F1@0.5 as our evaluation metric. The results are included in Table 2.

Table 2. Quantitative results from model experiments

Text	**Model**	**Parameter count**	**F1@0.5**
Recent LLMs (no fine-tuning)	OPT	1.3B	0.664
	Bloom	7B	0.621
	GPT3 - Curie	6.7B	0.718
	GPT3 - Babbage	13B	0.727
	GPT3 - Ada	2.7B	0.704
	Flan-T5-XL	3B	0.723
	Falcon	7B	0.745
	Llama	7B	0.751
BERT models (only fine tuning)	BERT	110M	0.729
	DistilBERT	66M	0.725
	RoBERTa	125M	0.728
	XLM-RoBERTa	125M	0.731
	Albert	12M	0.704
Fine-tuned LLMs & our custom architecture	OPT	1.3B	0.673
	Bloom	7B	0.643
	GPT3 – Curie	6.7B	0.727
	GPT3 – Babbage	13B	0.736
	GPT3 – Ada	2.7B	0.716
	SAS-BERT (A)	**110M**	**0.751**
	SAS-BERT (B)	**110M**	**0.762**
	Falcon	**7B**	**0.761**
	Llama	**7B**	**0.766**

As can be seen from the table, LLMs that were not fine-tuned for domain-specific email classification did not perform very well on this task, demonstrating a performance equivalent to simpler fine-tuned BERT family of models. The performance of Llama [4] and Falcon [3], without fine-tuning was the best of all LLMs considered. On fine-tuning LLMs, the performance of these models experienced some improvements, especially that of Llama and Falcon. To note, Llama and Falcon fine-tuning was performed using LoRA. This aligns with industry observations of the strength of these two open-source models. Surprisingly, the GPT-3 family of models exhibit only a marginal improvement in performance on fine-tuning. Our custom architecture shown in the table, SAS-BERT (A), demonstrates significant improvements compared to its simpler fine-tuned only BERT version. This refers to the Naïve-Bayes token along with span masking for MLM + next email prediction (NEP) setup during the pre-training task and fine-tuned for classification without GPT data augmentation. With GPT-3.5 data augmentation, as in SAS-BERT (B), we see a further lift in performance. This observation is very promising and aligns with extensive research that has proven that obtaining clean datasets goes much further in harnessing better performance out of these models, than simply scaling model parameters.

We compared the performance lift of SAS-BERT (A) over the simple fine-tuned BERT model across various industries present in the test dataset. We see that the largest gains were obtained in the SaaS (Software as a service) and e-commerce industries, which also form most pre-training data. Figure 4 highlights the performance gains obtained.

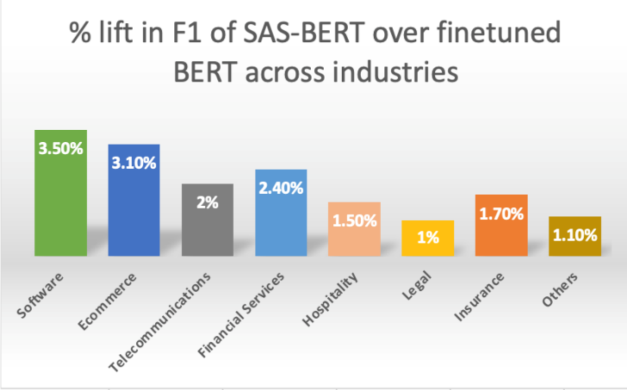
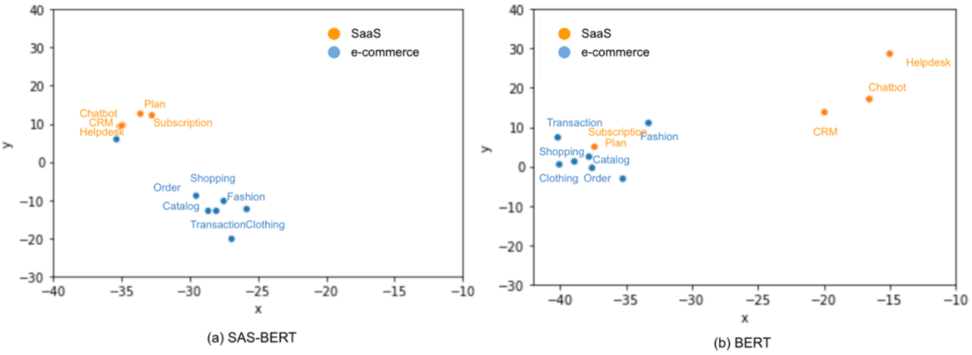


Figure 4. % lift in F1 of SAS-BERT over fine-tuned version of BERT across different industries.

Figure 5 demonstrates learned embeddings from our architecture in comparison to those from the fine-tuned only BERT model. As a result of pre-training with Naïve-Bayes token masking, keywords that are relevant to each industry appear closer together in the embedding space in our architecture. Since SaaS industry conversations form the majority in the datasets, embeddings learnt for these types of tokens seem to be much better and overall closer together in the embedding space. With more diverse data, there is a good promise of better embeddings that can be learnt across a wide array of businesses and industries.



(a) SAS-BERT

(b) BERT

Figure 5. t-SNE plots of embeddings related to common keywords in SaaS and e-commerce. (a) shows the t-SNE projections of these embeddings from our custom architecture and (b) for the same keywords from a fine-tuned only BERT model

5. Ablation Studies

We conducted detailed ablation studies to assess the impact of every enhancement included in the overall architecture. The results of the ablation studies are included in table 3.

The baseline for the ablation studies is an off-the-shelf BERT model fine-tuned using the training data described earlier. The result of this corresponds to the entry indexed type 0 in the table. In type 1, we undertake a simple pre-training of this BERT model architecture following the details included in the BERT paper [8].

Table 3: Detailed ablations study table demonstrating the impact of each customization and enhancement.

Type	BERT (fine-tuned only without GPT-3.5 augment.)	Pre-training without proposed changes	NB token masking for MLM	Masking span around NB token for MLM (NB-Span)	Next Email Prediction (NEP)	GPT-3.5 augment.	% F1 lift over baseline
0	✓						0.729
1	✓	✓					0.734
2	✓		✓				+1.6%
3	✓		✓	✓			+2.2%
4	✓				✓		+2.4%
5	✓		✓	✓	✓		+3.0%
6	✓					✓	+2.1%
7	✓		✓	✓	✓	✓	+4.5%

As can be seen, pre-training without any domain-specific modifications or smart customizations does not result in any appreciable improvement to the final F1 of the BERT model. The main challenge with vanilla pre-training seems to be the relatively small size of our pre-training data ~8M, whereas [20] and [17] that have reported noteworthy gains, on the back of pre-training with corpora of hundreds of millions of examples. Therefore, we are led to believe that smaller training datasets warrant special customizations to pre-training strategy in line with what we have experimented with.

Row indexed type 2 demonstrates an improvement of 1.6% over baseline through a mere replacement of random token masking with a more intelligent and context-driven Naïve Bayes token masking strategy. With type 3, we see that there is merit, although not very substantial, in forcing the model to predict a span of tokens around the NB token, with a total lift over baseline going up to 2.2%. Type 4 demonstrates the surprising improvement of 2.4% in F1 through next email prediction (NEP) alone. Our hypothesis for this is that in sales and support conversations, conversational exchange and the underlying dynamics and sentiment is more informative towards pre-training, rather than presence of certain context-indicative keywords in email. Type 5 corresponds to SAS-BERT (A), covering all the pre-training enhancements detailed so far.

In type 6, we can clearly see the value of augmenting fine-tuning using GPT-3.5 created small corpus of clean conversations data. Just augmentation alone, without any pretraining provides a performance lift of 2.1%.

This observation is highly encouraging and opens doors for further discussions and avenues where LLMs can be cheaply used to improve the quality of domain-specific internal datasets, rather than the more ambitious pursuit of fine-tuning these large models on noisy data. Finally, type 7 is SAS-BERT (B), resulting in an overall lift of 4.5%, which is highly significant at our scale, coming closer to performances demonstrated by some of the fine-tuned LLMs like Llama and Falcon, without any of the cost, inference, and interpretability constraints of LLMs.

6. Limitations of Our Work

Our proposed architecture has been pre-trained, fine-tuned, and tested only on internally available sales and support conversations data spanning several years. Such an equivalent dataset is not available publicly to the best of our knowledge, especially those that also carry a binary target label, and thus is one of the limitations of our work. We plan to address this limitation, in part, by pre-training our architecture on our internal data, but fine-tuning it on public datasets coming from similar domains, and for tasks like intent detection and named entity recognition. This will enable us to test beyond labelled, conversational datasets. Additionally, email exchanges are not strictly multi-turn in nature and often lack coherence and continuity in agent query and prospect response scheme which, for instance, is seen in chats. Agents sending multiple emails before getting a reply from a customer is one such example. This lack of coherence is not addressed in the next email prediction task now, leading to a small percentage of email pairs lacking logical meaning in the dataset for next email prediction task.

7. Conclusion & Future Work

In this paper, we explained the limitations of LLMs with respect to their out-of-the-box performance, cost, and maintenance overheads among others, and proposed that for highly domain-specific tasks, BERT family of models are still a promising alternative. We then presented our custom BERT architecture for sales and support conversations, carefully detailing the architectural customizations implemented and tested. Furthermore, we showed that we can leverage the power of generative abilities of GPT to develop a small, clean dataset that can additionally be trained on, thereby resulting in an incremental performance boost. We demonstrated how these enhancements lead to marked improvements over and above that of the vanilla BERT model, bringing the architecture performance closer to the best performing fine-tuned LLMs. In the future, we plan to utilize this architecture for chat conversations and voice data, which we believe carry the same conversational dynamics. We also aim to utilize much bigger conversations datasets (>100M emails, support queries etc.) as we believe we still have room to extract more out of the architecture in the pre-training and fine-tuning steps. Finally, we plan to develop specialized customer-specific models, training only on their business specific data and thus harnessing better prediction performance than a global model.

REFERENCES

[1] OpenAI (2023) GPT4 Large language model

[2] Brown, Tom, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D. Kaplan, Prafulla Dhariwal, Arvind Neelakantan et al. "Language models are few-shot learners." Advances in neural information processing systems 33 (2020): 1877-1901

[3] Almazrouei, Ebtesam, Hamza, Alobeidli, Abdulaziz, Alshamsi, Alessandro, Cappelli, Ruxandra, Cojocaru, Merouane, Debbah, Etienne, Goffinet, Daniel, Heslow, Julien, Launay, Quentin, Malartic, Badreddine, Noune, Baptiste, Pannier, and Guilherme, Penedo. "Falcon-40B: an open large language model with state-of-the-art performance” (2023)

[4] Touvron, Hugo, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière et al. "Llama: Open and efficient foundation language models." arXiv:2302.13971 (2023

[5] Wu, Zihao, Lu Zhang, Chao Cao, Xiaowei Yu, Haixing Dai, Chong Ma, Zhengliang Liu et al. "Exploring the trade-offs: Unified large language models vs local fine-tuned models for highly-specific radiology nli task." arXiv preprint arXiv:2304.09138 (2023)

[6] Ding, Ning, Yujia Qin, Guang Yang, Fuchao Wei, Zonghan Yang, Yusheng Su, Shengding Hu et al. "Parameter-efficient fine-tuning of large-scale pre-trained language models." Nature Machine Intelligence 5, no. 3 (2023): 220-235

[7] Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. "Attention is all you need." Advances in neural information processing systems 30 (2017)

[8] Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. "Bert: Pre-training of deep bidirectional transformers for language understanding.” NAACL (2018)

[9] Dai, Haixing, Zhengliang Liu, Wenxiong Liao, Xiaoke Huang, Zihao Wu, Lin Zhao, Wei Liu et al. "Chataug: Leveraging chatgpt for text data augmentation." arXiv preprint arXiv:2302.13007 (2023)

[10] Arefyev, Nikolay, Dmitrii Kharchev, and Artem Shelmanov. "Nb-mlm: Efficient domain adaptation of masked language models for sentiment analysis." In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pp. 9114-9124. 2021

[11] Joshi, Mandar, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. "Spanbert: Improving pre-training by representing and predicting spans." Transactions of the association for computational linguistics 8 (2020): 64-77

[12] Liu, Yinhan, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and VeselinStoyanov. "Roberta: A robustly optimized bert pretraining approach." arXiv preprint arXiv:1907.11692 (2019).

[13] Guillaume Lample and Alexis Conneau. 2019. Crosslingual language model pretraining. Advances in Neural Information Processing Systems (NIPS).

[14] Yang, Zhilin, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R. Salakhutdinov, and Quoc V. Le. "Xlnet: Generalized autoregressive pretraining for language understanding." Advances in neural information processing systems 32 (2019).

[15] Beltagy, Iz, Kyle Lo, and Arman Cohan. "SciBERT: A pretrained language model for scientific text." 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 3615– 3620

[16] Gu, Yu, Robert Tinn, Hao Cheng, Michael Lucas, Naoto Usuyama, Xiaodong Liu, Tristan Naumann, Jianfeng Gao, and Hoifung Poon. "Domain-specific language model pretraining for biomedical natural language processing." ACM Transactions on Computing for Healthcare (HEALTH) 3, no. 1 (2021): 1-23

[17] Lee, Jinhyuk, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. "BioBERT: a pre-trained biomedical language representation model for biomedical text mining." Bioinformatics 36, no. 4 (2020): 1234-1240

[18] Liu, Zihan, Feijun Jiang, Yuxiang Hu, Chen Shi, and Pascale Fung. "NER-BERT: a pre-trained model for low-resource entity tagging." CoRR, abs/2112.00405 (2021)

[19] Chalkidis, Ilias, Manos Fergadiotis, ProdromosMalakasiotis, Nikolaos Aletras, and Ion Androutsopoulos. "LEGAL-BERT: The muppets straight out of law school." arXiv preprint arXiv:2010.02559 (2020).

[20] Araci, Dogu. "Finbert: Financial sentiment analysis with pre-trained language models." arXivpreprint arXiv:1908.10063 (2019)

[21] Chandra, Chetan, Xiao Jing, Mayank V. Bendarkar, Kshitij Sawant, Lidya Elias, Michelle Kirby, and Dimitri N. Mavris. "Aviation-BERT: A Preliminary Aviation-Specific Natural Language Model." In AIAA AVIATION 2023 Forum, p. 3436. 2023.

[22] Pedersen, JannikSkyttegaard, Martin SundahlLaursen, Pernille Just Vinholt, and ThiusiusRajeethSavarimuthu. "MeDa-BERT: A medical Danish pretrained transformer model." In The 24rd Nordic Conference on Computational Linguistics. 2023.

[23] Bressem, Keno K., Jens-Michalis Papaioannou, Paul Grundmann, Florian Borchert, Lisa C. Adams, Leonhard Liu, Felix Busch et al. "Medbert. de: A comprehensive germanbert model for the medical domain." Expert Systems with Applications 237 (2024): 121598.

[24] Tai, Wen, H. T. Kung, Xin Luna Dong, Marcus Comiter, and Chang-Fu Kuo. "exBERT: Extending pre-trained models with domain-specific vocabulary under constrained training resources." In Findings of the Association for Computational Linguistics: EMNLP 2020, pp. 1433-1439. (2020)

[25] Cui, Yiming, Wanxiang Che, Ting Liu, Bing Qin, and Ziqing Yang. "Pre-training with whole word masking for chinesebert." IEEE/ACM Transactions on Audio, Speech, and Language Processing 29 (2021): 3504-3514.

[26] Sanh, Victor, Lysandre Debut, Julien Chaumond, and Thomas Wolf. "DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter." (2019)

[27] Lan, Zhenzhong, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. "Albert: A lite bert for self-supervised learning of language representations." (2019)

[28] Wang, Wei, Bin Bi, Ming Yan, Chen Wu, Zuyi Bao, Jiangnan Xia, Liwei Peng, and Luo Si. "Structbert: Incorporating language structures into pre-training for deep language understanding." (2019).

[29] Jiang, Zi-Hang, Weihao Yu, Daquan Zhou, Yunpeng Chen, Jiashi Feng, and Shuicheng Yan. "Convbert: Improving bert with span-based dynamic convolution." Advances in Neural Information Processing Systems 33 (2020): 12837-1284

[30] Wu, Chien-Sheng, Steven Hoi, Richard Socher, and Caiming Xiong. "TOD-BERT: Pre-trained natural language understanding for task-oriented dialogue." EMNLP (2020)

[31] Wang, Peiyao, Joyce Fang, and Julia Reinspach. "CS-BERT: a pretrained model for customer service dialogues." In Proceedings of the 3rd Workshop on Natural Language Processing for Conversational AI, pp. 130-142. (2021)

[32] Hu, Edward J., Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. "Lora: Low-rank adaptation of large language models." arXiv:2106.09685 (2021)

AUTHORS

Aanchal Varma works as a senior data scientistatFreshworks. She has about 4 years of experience solving complex problems and building scalable solutions in the field of NLP, deep learning, language modelling and machine learning. Prior to Freshworks, Aanchal has worked with Samsung R&D Delhi on various speech and NLP based solutions.

Chetan Bhat is a Senior Staff Data Scientist at Freshworks, leading several data science and machine learning initiatives for Freddy Freshsales. Prior to joining Freshworks, Chetan worked at start-ups like Practo, solving grass-root level healthcare problems using tech and data, and also at large B2B SaaS companies like LogMein, in areas like fraud detection, feature recommendations and subscription churn prediction.