Knowledge Distillation of Domain-adapted LLMs for Question-Answering in Telecom

Anonymous ACL submission

Abstract

Knowledge Distillation (KD) is one of the approaches to reduce the size of Large Language Models (LLMs). A LLM with smaller number of model parameters (student) is trained to mimic the performance of a LLM of a larger size (teacher model) on a specific task. For domain-specific tasks, it is not clear if teacher or student model, or both, must be considered for domain adaptation. In this work, we study this problem from perspective of telecom domain Question-Answering (QA) task. We systematically experiment with Supervised Fine-tuning (SFT) of teacher only, SFT of student only and SFT of both prior to KD. We design experiments to study the impact of vocabulary (same and different) and KD algorithms (vanilla KD and Dual Space KD, DSKD) on the distilled model. Multi-faceted evaluation of the distillation using 14 different metrics (N-gram, embedding and LLM-based metrics) is considered. Experimental results show that SFT of teacher improves performance of distilled model when both models have same vocabulary, irrespective of algorithm and metrics. Overall, SFT of both teacher and student results in better performance across all metrics, although the statistical significance of the same depends on the vocabulary of the teacher models.

1 Introduction

Large Language Models (LLMs) are complex models that perform a wide range of tasks, while Small Language Models (SLMs) have fewer parameters and are more suited for specific, resource-constrained applications. It has been well established that domain adaptation improves performance of LLMs in technical domains, such as telecom (Soman and Ranjani, 2023; Bariah et al., 2023; Roychowdhury et al., 2024; Karapantelakis et al., 2024; Zou et al., 2024). The need for SLMs arises due to their efficiency and cost-effectiveness as against LLMs (Piovesan et al., 2024; Maatouk et al., 2024; Schick and Schütze, 2020). Techniques to reduce the size of LLMs while retaining much of their performance is an area of active research. Popular techniques include quantization (Zhang et al., 2023), pruning (Ma et al., 2023) and Knowledge Distillation (KD) (Gou et al., 2021).

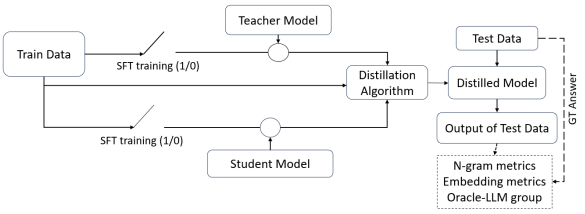

Figure 1: A schematic representation of experiments consisting of the choice of SFT for teacher student, the choice of distillation algorithms, Vanilla or DSKD, and choice of evaluation metrics.

In this work, we focus on the impact of domain adaptation of LLMs via KD approach. KD is a technique where a “student" (smaller) model is trained to replicate the performance of a “teacher" (larger) model (Gou et al., 2021; Xu et al., 2024) for a particular task. KD was originally proposed to reduce model size while retaining performance (Hinton et al., 2015).

1.1 Problem statement

The smaller models obtained during KD is said to improve generalization, reduce overfitting, especially when trained on small datasets. These small models enable faster inference and lower deployment cost. In domain specific tasks, such as telecom, it is important to ensure the models considered are domain aware. This is typically achieved through pre-training and/or supervised fine-tuning (SFT). SFT is a model training technique where a pre-trained model is further trained on a labeled dataset via supervised learning (Vaswani et al., 2017). To the best of our knowledge, there has been

no work pertaining to impact of domain adaptation through SFT of either teacher or student models prior to distillation. In addition, there are no insights on how one must choose the teacher and student models *viz.* must they be of same vocabulary or different. Lastly, quantifying performance of distilled generative models requires a holistic evaluation (Roychowdhury et al., 2024) as considering just N-gram metrics or embedding based similarity metrics can be severely limiting for LLMs. Hence, in-lieu of these gaps, we formalize the research questions in this work as follows:

- **RQ1:** Does SFT of teacher and/or student models prior to KD improve distilled model performance?
- **RQ2:** Does the choice of models for SFT and KD impact performance i.e., are there advantages in using models of different vocabulary over same vocabulary?
- **RQ3:** Does performance change for different metric groups - N-gram based, embedding based and Oracle-LLM based metrics?

1.2 Overview of KD techniques

Vocabulary of the models chosen for KD impact the performance of the distilled model. In this subsection, we give a quick overview of vocabulary prior to overview of KD techniques in literature.

1.2.1 Vocabulary

LLM vocabulary refers to the set of tokens (comprising of words, sub-words, or characters) that is used to represent text (Kolesnikova et al., 2022). Tokenization is the process of converting input text into tokens, and subsequently converted to embedding vectors. Tokenization and thus vocabulary plays an important role in model performance. It is evident that the vocabularies across LLM families differ based on the tokenization (Kolesnikova et al., 2022).

1.2.2 KD Algorithms

Typical KD training process involves comparing the token representations of the teacher and student models to align the latter to that of former using KL divergence (KLD) loss (Aguilar et al., 2020). We refer to this technique, in this paper, as vanilla KD.

When teacher and student model have different vocabulary, the models predict the tokens of next word in sequence from different vocabularies (Kolesnikova et al., 2022). Thus, it can be expected that vocabulary plays an important role in distilled model performance. Performance improvement in the scenario where the vocabularies do not match has been addressed by extending the vanilla KD approach by projecting each model's embeddings to a unified space; this is referred to Dual Space KD or DSKD (Zhang et al., 2024). For creating the unified space, the outputs from teacher model space are projected on to student model space and vice-versa. The transformation for projection is learnt during Cross-Model Attention (CMA) (Zhang et al., 2024) mechanism which also bridges the difference in vocabularies.

In this work, we consider two models (Llama family and Mistral family) and two KD approaches - vanilla KD and DSKD - to study the impact of vocabulary and algorithms on domain adaptation of LLMs. To the best of our knowledge, our work is the first to study the effect of domain adaptation through SFT of both the teacher and the student LLMs prior to distillation.

1.3 Overview of metrics

Evaluation of generated output from a LLM is an evolving research topic (Desmond et al., 2024; Roychowdhury et al., 2024). The current KD approaches typically report on few N-gram based metrics only (Zhang et al., 2024). For a more rounded evaluation of LLM, we consider three metric groups: N-gram based metrics, embedding based metrics and Oracle-LLM based metrics. The specific metrics are listed below:

- **N-gram based metrics**: BLEU, BLEU-CN , BLEU-DM , BLEU-DC (Shi et al., 2022), ROUGE-L Precision and Recall (Lin, 2004). These scores are indicative of overlap of N-gram word sequences or longest common sequences.
- **Embedding based metrics**: Cosine similarity (using all-Mini-L6-v2 embeddings (Reimers and Gurevych, 2020)), BERTScore (Zhang et al., 2019). These scores are indicative of semantic similarity.
- **Oracle-LLM based metrics**: RAG Assessment metrics (RAGAs) that uses Oracle-LLM to arrive at metrics such as faithfulness, factual correctness, answer similarity, answer correctness, answer relevance and context rele-

vance, (Es et al., 2024), (Roychowdhury et al., 2024).

Higher scores imply better model performance for all the metrics considered above. Capturing KD performance using set of metrics which cover word/token overlap, semantic similarity and generation perspective aids towards holistic analysis.

1.4 Contributions

From the experiments designed and through the results on TeleQuAD (Gebre et al., 2025), the contributions of this work are:

- This is the first work which addresses the effect of supervised fine tuning (SFT) of both the teacher and the student language models prior to distillation (with a focus on telecom domain QA task).
- We demonstrate that SFT of teacher and student models improves performance, irrespective of vocabulary and algorithm choice.
- We demonstrate SFT of teacher has significant performance improvements (across metrics) when using same vocabulary models.
- In scenarios where SFT training has practical limitations, using different vocabulary with DSKD algorithm is found to be useful.
- All group-wise metrics show similar performance trends.

The rest of the paper is organized as follows. The experimental design and evaluation metrics are described in Section 2, followed by experimental setup and results in Section 3. We conclude and discuss future work in Section 4.

2 Methodology

We describe the experimental setup, statistical tests and the details of dataset and models.

2.1 Experimental Setup

We describe the experimental design considered to study the impact of vocabulary (same and different) and KD algorithms (vanilla KD and Dual Space KD, DSKD) on the distilled model. We also analyze the impact of untrained/SFT teacher/student model on the final distilled model. Fig. 1 shows the schematic representation of our experimental setup. Depending on choice of SFT training, teacher model and distillation algorithms, there are 4 parameters of interest here:

Notation	Description
$T_B(L)$	Base Llama as teacher model (Same vocabulary)
$T_{SFT}(L)$	SFT Llama as teacher model (Same vocabulary)
$T_B(M)$	Base Mistral as teacher model (Different vocabulary)
$T_{SFT}(M)$	SFT Mistral as teacher model (Different vocabulary)
S_B	Base TinyLlama as student model
S_{SFT}	SFT TinyLlama as student model
V and D	Vanilla algorithm and DSKD algorithm for KD process

Table 1: A summary of notations used to formulate hypothesis tests and report results.

- **Teacher** – Two variants of the teacher model, base model and SFT model, arising out of SFT training (depicted as a 1/0 switch) in Fig. 1.
- **Student** – Similarly, student model is also considered with two variants - base model and SFT model (refer to Fig. 1). These two addresses RQ1.
- **Vocabulary** – To study impact of vocabulary, we consider two cases where teacher and student (i) both having same vocabulary (ii) both having different vocabulary. We fix student model to be from the Llama family - TinyLlama. Hence, with respect to the student SLM TinyLlama, choosing the teacher model as (i) LLM Llama results in same vocabulary (ii) LLM Mistral results in different vocabulary. This addresses RQ2.
- **KD algorithm** – To analyze if insights on vocabulary is invariant to choice of KD algorithm, we consider two KD algorithms - Vanilla KD and DSKD.

Fig. 2 shows a schematic representation of 16 combinations of experiments arising out of the design described above. The notations used to depict the various combinations are summarized in Table 1. We report performance using 14 metrics (refer Section 1.3) for each of the 16 combinations of distillation experiments.

2.2 Hypothesis tests

In addition to reporting the performance metrics, we analyze the impact of the SFT of teacher, stu-

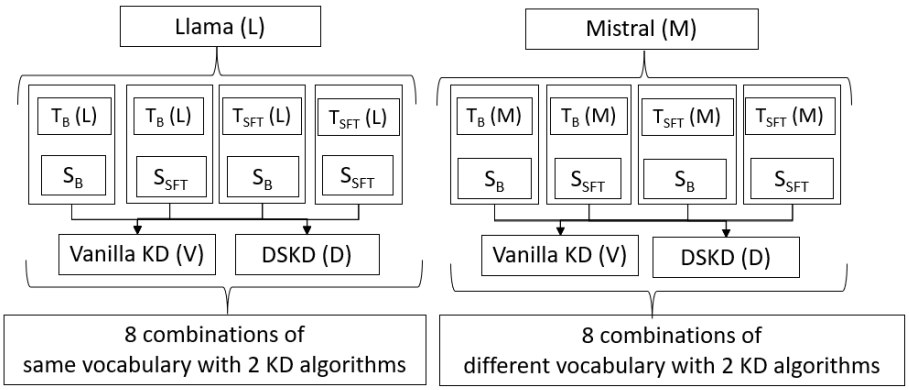

Figure 2: Schematic representation of different choices based on which we conduct Hypothesis tests

dent, model vocabulary and the algorithm chosen for each metric. This results in 16 combinations of results for 14 metrics (RQ3); to ensure the results are statistically significant, we group the results to perform statistical hypothesis tests (Wilcoxon statistics signed rank test (Gehan, 1965)).

We henceforth use the notation where a tuple (T_B, S_B) indicates a teacher-student pair where the models are identified by the acronyms as in Table 1. Using a wildcard * in the suffixes indicate all possible options of the latter. When we use a third term in the tuple e.g. D or V we refer to the corresponding algorithm - not having this indicates that we test for both algorithms. We compare each of the 14 metrics via a statistical test and this is indicated by the $Perf()$ function.

- **H-Train**: We consider the null hypotheses (NH), Eq. (1), to analyze if T_{SFT} or S_{SFT} or both (followed by KD) impacts the performance of the distilled model (RQ1).

$$H^{T}_{train} : Perf(T_B, S_B) = Perf(T_{SFT}, S_B)$$
$$H^{S}_{train} : Perf(T_B, S_B) = Perf(T_B, S_{SFT})$$
$$H^{T,S}_{train} : Perf(T_B, S_B) = Perf(T_{SFT}, S_{SFT}) \quad (1)$$

  The alternate hypotheses to all of these correspond to $Perf(T_B, S_B) \neq Perf(T_*, S_*)$ respectively where $*$ corresponds to the SFT of teacher or student or both. For each of the three NH above, impact on vocabulary (L and M) and algorithm (V and D) choice are also considered. So, we have 12 hypothesis tests for each of the 14 metrics.

- **H-SFT**: Results show that T_{SFT}, S_{SFT} combination results in best performance across metrics (discussed later in Section 3). To analyze the impact of the T_{SFT} only, S_{SFT} only and (T_{SFT}, S_{SFT}) prior to the distillation process (RQ1), we formulate NH as Eq. (2).

$$H^{T}_{SFT} : Perf(T_{SFT}, S_{SFT}) = Perf(T_{SFT}, S_B)$$
$$H^{S}_{SFT} : Perf(T_{SFT}, S_{SFT}) = Perf(T_B, S_{SFT}) \quad (2)$$

  Again, the alternate hypotheses to all of these correspond to $Perf(T_{SFT}, S_{SFT}) \neq Perf(T_*, S_*)$ respectively. Here, $*$ refers to SFT of teacher model only or student model only. Each of the binary choice of vocabulary (M, L) and algorithm (V, D) is considered resulting in 8 tests for each of the 14 metrics.

- **H-Algo**: Impact of KD algorithm (RQ2) post SFT through NH is shown in Eq. (3).

$$H^{T,S}_{Alg} : Perf(T_{SFT}, S_{SFT}, V) = Perf(T_{SFT}, S_{SFT}, D)$$
$$H^{B}_{Alg} : Perf(T_B, S_B, V) = Perf(T_B, S_B, D) \quad (3)$$

  This is considered for both the vocabulary scenarios, (M, L), and scenario of best performing SFT models (T_{SFT}, S_{SFT}) and untrained models (T_B, S_B); the latter accounts for scenarios when training models is not feasible. This results in 4 NH tests for each of the 14 metrics.

2.3 Dataset description

We consider samples from TeleQuAD (Gebre et al., 2025), a Telecom QA dataset, curated using publicly available 3GPP (Rel 15) documents (3GPP, 2019). The training, development and test data comprise of 2385, 726 and 597 QA pairs, respectively, derived from 452 contexts (sections).

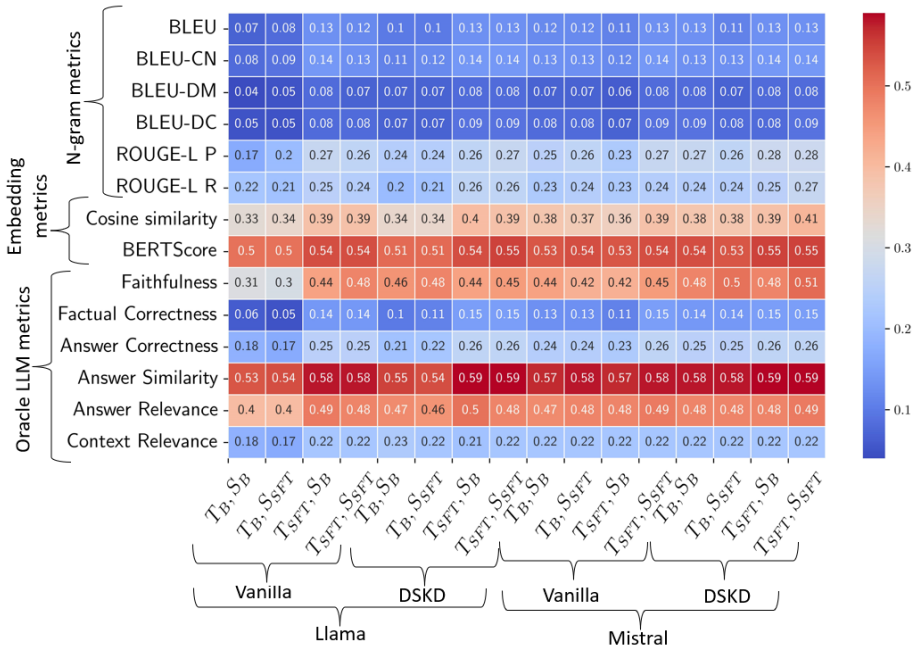

Figure 3: Performance on 14 metrics for various combinations of T_B, T_{SFT}, S_B, S_{SFT} using two KD algorithms (Vanilla and DSKD) and models of different and same vocabulary (Mistral and Llama).

2.4 Environment

In our experiments, we have considered Llama-7b[1], Mistral-7b[2] as teacher models and Tinyllama-1.1b[3] as student model. The GPU used for training and inference is NVIDIA A100-SXM4-80GB. Table 2 summarizes the parameters considered for T_{SFT} and S_{SFT}.

Model source	Huggingface model hub[4]
Maximum epoch	50
Early stopping criteria	minimum improvement + 0.01
Early stopping patience	3 epochs
Learning rate	0.001, cosine decay
SFT algorithm	Low-Rank Adaptation (LoRA) (Hu et al., 2021)
Rank	256
Alpha	8
Dropout	0.1

Table 2: Summary of the parameters for SFT.

3 Experimental Results

Fig. 3 shows the heatmap depicting performance of 16 combinations of KD for 14 metrics. For brevity, we also report the mean of all 14 metrics and group-wise metrics (N-gram metrics, embedding based metrics and Oracle-LLM metrics) in Fig. 4.

We systematically analyze the results and organize our findings as impact of (i) SFT (RQ1) (ii) SFT on teacher and student (RQ1) (iii) vocabulary and KD algorithm (RQ2) (iv) performance metrics groups (RQ3)

3.1 Impact of SFT

We organize analysis with vocabulary as starting point:

3.1.1 Llama

Consider the bar plots which depicts Llama as the teacher in Fig. 4 i.e., the bars denoting (Llama, Vanilla KD) and (Llama, DSKD). We observe that SFT of teacher/student/both results in improvement of performance irrespective of the training algorithm (first bar vs the subsequent 3 bars). The improvement is statistically significant (refer to H^S_{train}, H^T_{train}, $H^{T,S}_{train}$ in Table 3). Here, we observe that NH is rejected for most metrics (13 out of 14 for Vanilla KD and 8 or 9 out of 14 for DSKD) with SFT of student or teacher or both for Llama vocabulary, irrespective of algorithms. From this and from the average performance metrics, we infer that SFT results in statistically significant performance improvement when we choose models of same vocabulary (Llama and TinyLlama pair) mod-

[1] https://huggingface.co/meta-llama/Llama-2-7b
[2] https://huggingface.co/mistralai/Mistral-7B-v0.1
[3] https://huggingface.co/TinyLlama/TinyLlama-1.1B-Chat-v1.0

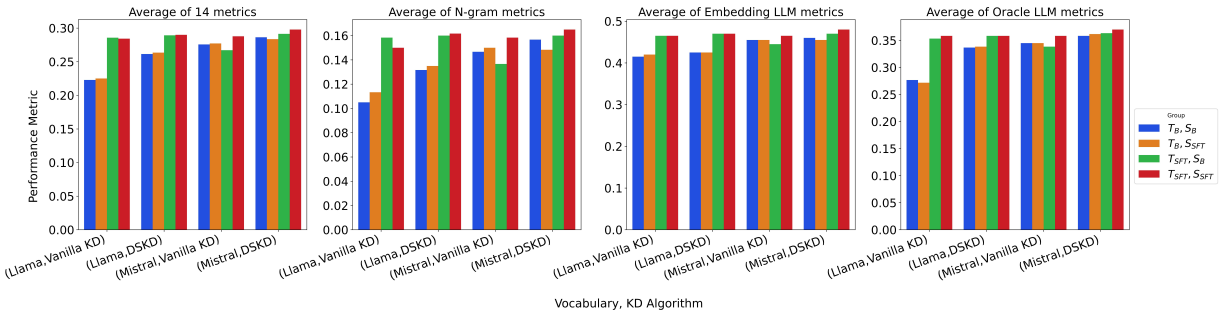

Figure 4: Group-wise average of performance metrics from the heatmap in Fig. **??**.

els.

3.1.2 Mistral

Refer to Mistral set of bar plots in Fig 4 i.e., the bars denoting (Mistral, Vanilla KD) and (Mistral, DSKD). We observe that training improves results Fig. 4, but the improvement is not statistically significant (refer row Mistral in Table 3); NH is not rejected H^S_{train}, H^T_{train}, $H^{T,S}_{train}$ i.e., 0 out of 14 metrics are NH rejected.

Thus, combining these findings, we infer that **training using SFT improves performance across vocabulary and algorithms; improvement with SFT of teacher and/or student when models have same vocabulary is significant**.

3.2 Impact of SFT on teacher and student

3.2.1 Llama

We observe from Llama set of bar plots of Fig 4 (i.e., the bars denoting (Llama, Vanilla KD) and (Llama, DSKD)), that SFT results in improved performance for (T_{SFT}, S_B) and (T_{SFT}, S_{SFT}) than for (T_B, S_{SFT}) across metric groups. From Table 3, Llama row and H^S_{SFT}, we see that NH is rejected for most cases irrespective of the KD algorithm - improvement of (T_{SFT}, S_{SFT}) over (T_B, S_{SFT}) is more significant than that of (T_{SFT}, S_{SFT}) over (T_{SFT}, S_B) combination i.e., NH is rejected in 13 and 9 out 14 metrics for H^S_{SFT} and not rejected for any metric in H^T_{SFT}. This implies that **SFT of teacher model before KD is useful and it is not necessary to train both teacher and student when choosing models of same vocabulary**.

3.2.2 Mistral

When vocabulary is different i.e., refer Mistral set of results in Fig 4, the bars denoting (Mistral, Vanilla KD) and (Mistral, DSKD), we observe that best performance seen in (T_{SFT}, S_{SFT}), followed by (T_B, S_{SFT}), and an apparent dip in performance

Algorithm	Llama-V	Llama-D	Mistral-V	Mistral-D
H^S_{train}	0	1	0	0
H^T_{train}	13	9	0	0
$H^{T,S}_{train}$	13	8	0	0
H^S_{SFT}	13	9	0	0
H^T_{SFT}	0	0	0	0
$H^{T,S}_{Alg}$	0		1	
H^B_{Alg}	10		14	

Table 3: Count of metrics for which NH is rejected for each of the hypotheses listed in Section 2.2

is observed for (T_{SFT}, S_B) or (T_B, S_{SFT}). Referring to Table 3 for Mistral, all the H-SFT tests show that NH is not rejected for any of the metrics for both H^T_{SFT} and H^S_{SFT}. This implies **the performance improvement/dip with both models trained or either model trained is not statistically significant irrespective of the KD algorithm**. We suspect this could be because of limited training data and one of the potential future work direction could be towards SFT results with more training samples.

3.3 Impact of KD algorithm and Vocabulary

3.3.1 Algorithm

Consider the best performing pair of models i.e., (T_{SFT}, S_{SFT}). We observe that the model performance improvement exists, but is not significant enough; refer to Table 3 where NH is not rejected for most of the metrics for $H^{T,S}_{Alg}$, across vocabularies (NH rejected is 0 out of 14 for Llama and 1 out of 14 metrics for Mistral model). **This implies, KD performance doesn't depend on the algorithm choice with SFT of both teacher and student. However, when SFT is not feasible, we observe that performance is (statistically) better with DSKD algorithm** because NH is rejected for most of the metrics for (10 out of 14 and 14 out of

14) vocabulary with H^B_{Alg}.

3.3.2 Vocabulary

Consider barplots corresponding to (T_{SFT}, S_{SFT}) in Fig. 4, i.e., SFT of both teacher and student models. We observe that average of all of the metrics are similar. Hence, **vocabulary does not impact performance with both models are trained.** However, **when training is not a feasibility, using models from different vocabulary with DSKD algorithm shows better performance.**

3.4 Impact of Performance Metrics

From Fig. 4 and Table 3, we observe that although the range of results of the 3 groups of metrics are different, the trends followed across groups are in alignment. **There is no metric group where the results are contradictory.**

3.5 Summary

We summarize the findings from the results section above here.

- RQ1
  - Training teacher models, student models or both using SFT improves performance across Llama, Mistral models and Vanilla KD and DSKD algorithms.
  - When teacher and student models are Llama and tinyLlama, SFT of teacher before KD is useful and it may not be necessary to train both teacher and student.
  - When teacher model is Mistral, and student model is TinyLlama, the performance improvement with SFT of both teacher and student models is not statistically significant over that of either model being trained, and this holds irrespective of the KD algorithm.
- RQ2
  - If one performs SFT of both teacher and student, distilled model performance doesn't depend on the algorithm or teacher model (Llama or Mistral) choice.
  - When SFT is not feasible, we observe that performance is (statistically) better using Mistral as teacher model, TinyLlama as student model with DSKD algorithm.
- RQ3
  - The performance results follow similar trends across metric groups.

4 Conclusions & Future Work

In this work, we have systematically studied impact of SFT of teacher and student model prior to KD of LLMs from perspective of vocabulary match, KD algorithms, variants of teacher SFT or student SFT or both. Our results are based on a telecom QA dataset and we use various metrics for an overall perspective. We have discussed outcome of RQs through performance results and statistical tests. From our analysis, we recommend that when teacher is Mistral, training using SFT improves performance across vocabulary and algorithms; improvement with SFT of teacher and/or student when models have same vocabulary is significant. When Llama is the teacher, SFT of teacher model before KD is useful and it is not necessary to train both teacher and student when choosing models of same vocabulary.

Future work would involve extending it to other tasks like code generation and agent-based systems. Another direction for future work is towards model size - the teacher models used in this work are of relatively smaller size. Evaluation of KD from larger models including Mixture of Experts (MoE) models for domain-specific tasks would be important for the community.

References

3GPP. 2019. 3GPP release 15. Technical report, 3GPP. Accessed: 2024-05-19.

Gustavo Aguilar, Yuan Ling, Yu Zhang, Benjamin Yao, Xing Fan, and Chenlei Guo. 2020. Knowledge distillation from internal representations. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 7350–7357.

Lina Bariah, Hang Zou, Qiyang Zhao, Belkacem Mouhouche, Faouzi Bader, and Merouane Debbah. 2023. Understanding telecom language through large language models. In *GLOBECOM 2023-2023 IEEE Global Communications Conference*, pages 6542–6547. IEEE.

Michael Desmond, Zahra Ashktorab, Qian Pan, Casey Dugan, and James M Johnson. 2024. Evalullm: Llm assisted evaluation of generative outputs. In *Companion Proceedings of the 29th International Conference on Intelligent User Interfaces*, pages 30–32.

Shahul Es, Jithin James, Luis Espinosa Anke, and Steven Schockaert. 2024. Ragas: Automated evaluation of retrieval augmented generation. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 150–158.

Fitsum Gebre, Henrik Holm, Maria Gunnarsson, Doumitrou Nimara, Jieqiang Wei, Vincent Huang, Avantika Sharma, and H G Ranjani. 2025. TeleQuAD: A suite of question answering datasets for the telecom domain.

Edmund A Gehan. 1965. A generalized wilcoxon test for comparing arbitrarily singly-censored samples. *Biometrika*, 52(1-2):203–224.

Jianping Gou, Baosheng Yu, Stephen J Maybank, and Dacheng Tao. 2021. Knowledge distillation: A survey. *International Journal of Computer Vision*, 129(6):1789–1819.

Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. 2015. Distilling the knowledge in a neural network. *ArXiv*, abs/1503.02531.

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

Athanasios Karapantelakis, Mukesh Thakur, Alexandros Nikou, Farnaz Moradi, Christian Olrog, Fitsum Gaim, Henrik Holm, Doumitrou Daniil Nimara, and Vincent Huang. 2024. Using large language models to understand telecom standards. In *2024 IEEE International Conference on Machine Learning for Communication and Networking (ICMLCN)*, pages 440–446.

Alina Kolesnikova, Yuri Kuratov, Vasily Konovalov, and Mikhail Burtsev. 2022. Knowledge distillation of russian language models with reduction of vocabulary. *arXiv preprint arXiv:2205.02340*.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Xinyin Ma, Gongfan Fang, and Xinchao Wang. 2023. Llm-pruner: On the structural pruning of large language models. *Advances in neural information processing systems*, 36:21702–21720.

Ali Maatouk, Nicola Piovesan, Fadhel Ayed, Antonio De Domenico, and Merouane Debbah. 2024. Large language models for telecom: Forthcoming impact on the industry. *IEEE Communications Magazine*.

Nicola Piovesan, Antonio De Domenico, and Fadhel Ayed. 2024. Telecom language models: Must they be large? *arXiv preprint arXiv:2403.04666*.

Nils Reimers and Iryna Gurevych. 2020. Making monolingual sentence embeddings multilingual using knowledge distillation. *arXiv preprint arXiv:2004.09813*.

Sujoy Roychowdhury, Sumit Soman, H. G. Ranjani, Neeraj Gunda, Vansh Chhabra, and Sai Krishna Bala. 2024. Evaluation of RAG metrics for question answering in the telecom domain. In *ICML 2024 Workshop on Foundation Models in the Wild*.

Timo Schick and Hinrich Schütze. 2020. It's not just size that matters: Small language models are also few-shot learners. *arXiv preprint arXiv:2009.07118*.

Ensheng Shi, Yanlin Wang, Lun Du, Junjie Chen, Shi Han, Hongyu Zhang, Dongmei Zhang, and Hongbin Sun. 2022. On the evaluation of neural code summarization. In *Proceedings of the 44th international conference on software engineering*, pages 1597–1608.

Sumit Soman and H. G. Ranjani. 2023. Observations on LLMs for telecom domain: capabilities and limitations. In *Proceedings of the Third International Conference on AI-ML Systems*, pages 1–5.

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

Xiaohan Xu, Ming Li, Chongyang Tao, Tao Shen, Reynold Cheng, Jinyang Li, Can Xu, Dacheng Tao, and Tianyi Zhou. 2024. A survey on knowledge distillation of large language models. *arXiv preprint arXiv:2402.13116*.

Cheng Zhang, Jianyi Cheng, Ilia Shumailov, George A Constantinides, and Yiren Zhao. 2023. Revisiting block-based quantisation: What is important for sub-8-bit llm inference? *arXiv preprint arXiv:2310.05079*.

Songming Zhang, Xue Zhang, Zengkui Sun, Yufeng Chen, and Jinan Xu. 2024. Dual-space knowledge distillation for large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing (EMNLP),Miami Florida USA*.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.

Hang Zou, Qiyang Zhao, Yu Tian, Lina Bariah, Faouzi Bader, Thierry Lestable, and Merouane Debbah. 2024. Telecomgpt: A framework to build telecom-specfic large language models. *arXiv preprint arXiv:2407.09424*.