THE EFFICIENCY PARADIGM: HOW SMALL LANGUAGE MODELS ARE FORGING A MENU-LESS, CLICK-LESS FUTURE FIOR ENTERPRISE WORKFLOW AUTOMATION

Vivek Pandey
MD, Bright Hustle , Bhopal

Abstract:

While LLMs have shown great generalist capabilities, difficulties like inference latency, high computation expenses and data privacy concerns with cloud-based APIs are limiting their use in enterprise administrative workflows. These restrictions make it difficult to adopt more widespread and efficient automation for more complex administrative tasks in specific domains. The authors suggest that the future of enterprise transformation lies in Small Language Models (SLMs), which are efficient and accurate, task-specific models. We believe that SLMs are not scaled down LLMS but a much better paradigm for enterprise applications. We present Devki, a new SLM tailored for enterprise workflows for administration. We explain its design, training process using knowledge distillation and Parameter-Efficient Fine-Tuning (PEFT), and its results on a variety of common administrative tasks. Our empirical evaluation shows that the accuracy, efficiency (latency and resource consumption), and cost-effectiveness of Devki over a general purpose LLM baseline are significant and meaningful for domain-specific tasks. We demonstrate how Devki is making possible and changing forever the nature of human-computer interaction in the enterprise with a "menu-less, click-less" conversational workflow. This book lays the groundwork for the strategic benefits of SLMs in the enterprise, offers a repeatable process for creating SLMs and brings a new and more effective age in enterprise technology to life with Devki.

I. INTRODUCTION

Today's business environment is comprised of a myriad of administrative processes. In recent years, this has been a major strategic goal for any organization in any industry, with powerful Artificial Intelligence (AI) – specifically the Natural Language Processing (NLP) – now on the horizon to transform the world of business.

Large Language Models (LLMs) like OpenAI's GPT-4 and Google's Gemini have become the center of attention for this AI revolution, captivating the world with their extensive knowledge and capacity to undertake numerous language-related tasks. But there's a contradiction when considering the use of these all-encompassing, general-purpose models in the enterprise landscape. In principle they could be practical, but in reality they will have their difficulties and will prove to be a significant obstacle in specific administrative settings. The amount of compute required for training and inference is also significant, requiring massive-scale cloud infrastructure and thousands of high-end GPUs – and the costs of inference are in the millions of dollars. This computational burden also means that LLM models are not well suited to the kinds of real-time, interactive applications that are central to modern workflows.

Most seriously, the main way people consume LLMs via an API is in the cloud, opening deep security and privacy worries. In addition, as LLMs are "generalists" trained on a vast range of information on the public internet, they are susceptible to factual inaccuracies, or "hallucinations," when exposed to the specific jargon and context of a specialized business area.

The philosophy of 'specialization and efficiency,' rather than ever larger, more general models, is the key to scalable, effective, and secure enterprise AI, this paper contends. We believe the future of administrative workflow

Small Language Models (SLMs) will be the key to defining automation. SLMs have been developed to directly address these limitations of LLMs and provide a better alternative in terms of performance and precision, capability and cost, and power and privacy.

The implications of this technological transformation go beyond simply the

automation of tasks; it will impact how humans interact with computers in the workplace. The GUI paradigm of clicks, icons and menus has been the reigning rule for decades. The next evolutionary step is the Conversational User Interface (CUI): complex multiple-step tasks are accomplished in a natural, fluid dialog, hiding all but the most basic technology complexities from the user. SLMs enable this "menu-less, click-less" user experience by directly embedding specialized linguistic intelligence into enterprise systems, making technology more accessible and intuitive than ever.

In order to prove this thesis, this paper presents Devki, a novel SLM designed, trained and evaluated specifically for the Enterprise Administrative Workflow (EAWf) domain. Our contributions in this work are four folds: First, we do a rigorous comparison of the architectural, performance and economic qualities of SLM vs LLM in an enterprise environment. Second, we will provide a clear and precise method for the construction of Devki, its architecture, the way to curate the set of data, and the tuning procedure used in fine-tuning Devki. Third, we give an empirical assessment of Devki against a top general-purpose LLM, showing that it outperforms this LLM on a representative set of administrative tasks. Last but not least, we present a useful example of Devki, showing how it brings the capabilities needed to implement a multi-step workflow without a menu as well as being a proof of concept of this new age of enterprise technology.

II. THE CASE FOR SPECIALIZATION: A COMPARATIVE ANALYSIS OF SLMS AND LLMS

Using SLMs in the enterprise isn't just about being cost-effective - it's about the fundamental differences in architecture, performance attributes and deployment viability. Scale, as introduced above, is a characteristic of LLMs, but not of SLMs. Efficiency and accuracy are the characteristics of This section discusses the benefits of the SLM paradigm systematically in the context of the corporate environment.

A. Architectural and Efficiency Advantages ofi SLMs

The scale of the difference is not only quantitative—with millions to the low billions (e.g., 1-13 billion) of parameters, in contrast to LLMs which can have hundreds of billions to trillions of parameters—but also allows a range of architectural innovations essential for the SLM value proposition.

Self-aflention is one of the most important parts of the transformer models that requires quadratic computational resources with the number of the sequence. SLMs overcome this boflleneck by greater efficient aflention variants. Other methods such as the Grouped-Query Attention (GQA) and Multi-Query Attention (MQA) will also decrease the memory usage of the key-value (KV) cache during inference, which consumes most of the memory while running the model.

Many LLMs (such as Mistral 7B) use the Sliding Window Attention (SWA) mechanism instead of the traditional self-aflention to process long contexts efficiently, offering a linear-time alternative to the self-aflention used in most LLMs.

The community is also actively investigating architectures that are extremely efficient, but not based on transformers, such as using shared Feed-Forward Networks (FFNs) or repeated blocks of layers. In addition to that, architectures that use lightweight components such as SwiGLU activation functions and RMSNorm for layer normalization are being used.

Lastly, there are some interesting alternatives such as State Space Models (SSMs) models as demonstrated by Mamba and other sequential models like RWKV, which are especially suitable for the design of high performance SLMs.

All of the architectural decisions made to achieve computational efficiency, have a cascading effect and directly impact enterprise needs. Immediate technical enablers for SLM deployment on consumer-grade hardware with lower memory and computation requirements are reduced memory footprint and lower computational demand than the massive energy-hungry clusters of computers needed by LLMs.

This feasibility of local deployment, whether on an organisation's own servers (on-premise) or directly on the edge devices, is the essential element to overcome the critical business challenges around data privacy, security and regulatory compliance. When sensitive data never needs to pass through the corporate firewall to be processed by a third party API, issues of data sovereignty and confidentiality are effectively neutralised.

B. Perfiormance in Domain-Specific Contexts

This is because there is a difference in architecture, training data, and thus, performance between LLM and SLM. Whereas LLM's tend to be generalists, often trained on large and diverse collections of data extracted from the public internet, they acquire a shallow knowledge of a wide range of topics, SLM's are specialists, typically fine-tuned on

smaller, high-quality and curated datasets that cover a specific domain, like legal contracts, financial reports, or internal IT support tickets.

It gives an important performance benefit in enterprise seflings. As with any other kind of text-based task, a fine-tuned SLM does a better job producing high-quality, accurate, highly structured, domain-specific output—such as a workflow in JSON when the user provides a request—for tasks that demand a degree of understanding and require highly structured outputs, such as a workflow, in a specific domain from a structured user request. The targeted training on domain-specific data has been shown to significantly improve performance on such tasks by 10% or more over that of a general purpose LLM that lacks any in-domain knowledge. The former's ability to "hallucinate" – to produce incorrect or nonsensical information – a type of failure common to many LLMs outside of their training set, is severely limited.

SLMs can sometimes have a smaller context window, but with a narrower context window tend to be better at handling the context in their specific domain. The precision is crucial in administrative workflows where accuracy cannot be an option – even for a large-scale model, a generalist one may miss such subtle dependencies and implicit knowledge in domain-specific language.

C. Economic and Deployment Viability

While there are many differences between SLM and LLM, the most apparent one is that on economic and deployment aspects, SLM is more convenient for the average enterprise as compared to LLM. The Total Cost of Ownership (TCO) of an SLM solution is of orders of magnitude less than that of a solution based on a large general purpose model.

This is also much more affordable, usually in the thousands of dollars as opposed to millions needed to train a frontier LLM from scratch, and becomes even more pronounced due to the ubiquity of Parameter-Efficient Fine-Tuning (PEFT) techniques.) The methods such as Low-Rank Adaptation (LoRA), Quantized LoRA (QLoRA) and other prompt tuning techniques enable the developers to modify the pre-trained SLM with only a small portion of its parameters, drastically cutting down on the amount of computation and time needed to customize the SLM.

The generation of responses is an on-going operational cost – Inference. As in many other areas, SLMs are very effective. The reduced size and optimized designs result in much lower latency and computational power, which is particularly valuable in real-time,

interactive applications, and also significantly decreases cloud or on-premise server recurring costs, which are increasingly gaining importance in relation to sustainability and environmental, social, and governance (ESG) goals within corporate business.

Lastly, the portability of SLMs is easy to deploy, but not that of most LLMs. They can be used on any server, from on-premises in a corporate data center, edge computing devices at a factory or retail store, or even directly on an employee's mobile phone or laptop. This has the potential to revolutionize industries that have strict data residency requirements, apps that need to be reliable without internet connection and any workflow where a low latency response matters to the user experience.

Characteristic	**Small Language Model (SLM)**	**Large Language Model (LLM)**
Parameter Size	Millions to low billions (e.g., < 20B)	Tens of billions to trillions
Training Data Scope	Narrow, curated, domain-specific	Broad, general, internet-scale
Training/Fine-Tuning Cost	Low (thousands of dollars), PEFT-friendly	High to very high (millions of dollars)
Inference Latency	Low, suitable for real-time applications	High, often unsuitable for interactive use
Resource Consumption	Low compute and energy requirements	Very high compute and energy requirements
Deployment Model	Flexible: Cloud, on-premise, edge, mobile	Primarily cloud-based via API
Data Privacy/Security	High (enables on-premise/local processing)	Lower (relies on third-party data processing)
Accuracy (General Tasks)	Moderate	High
Accuracy (Specialized Tasks)	High to very high (when fine-tuned)	Moderate, prone to domain-specific errors

Primary Use Case	Task-specific automation, specialized agents	General-purpose chatbots, broad content creation

Table I: A comparative analysis of the key characteristics distinguishing Small Language Models (SLMs) from Large Language Models (LLMs) in the context of enterprise applications. Data synthesized from.[9]

III. DEVKİ: A TASK-ORIENTED SLM FOR ENTERPRISE WORKFLOW AUTOMATION

In this section, the design, development and evaluation of a novel SLM, Devki, specifically engineered for enterprise administrative workflow automation is described, as a way to give empirical support to dedicated SLM paradigm. Devki is a real-world example of the principles described in the last section, in which architecture efficiency and domain knowledge are combined to give a powerful and useful tool to the modern enterprise.

A. Architecture and Design Principles ofi Devki

Devki is a 7-billion parameter model, using only a decoder. These features are all essential to achieve the low latency and low-resource use demands of enterprise deployment and were therefore chosen to be retained as the core architecture for Devki, based on the open-source Mistral 7B architecture. The model uses Grouped-Query Attention (GQA) that offers a good balance between the speed of Multi-Query Aflention and quality of Multi-Head Aflention. The above capabilities are specifically utilised by Devki, a company that processes a large number of administrative documents such as invoices or compliance reports every day.

It can accommodate longer contexts in documents (up to an 8,000-token window) by linear computation using a technique called Sliding Window Attention (SWA). For scenarios such as summarizing long legal contracts, for example, and analysing tedious service tickets without being overwhelmed by the quadratic scaling problems which are common with regular aflention mechanisms. A single design principle was used for Devki – optimise for the specific needs of administrative work. They are frequently text-based, need to adhere to a format, demand high factual accuracy and must be completed with little delay to interface with interactive user workflows.

The choice of a The 7B parameter model offers a balance between expressiveness and

efficiency, with enough capability to handle complex business logic, but size enough to be fine-tuned and deployed on the premises without high costs.

B. Training and Fine-Tuning Strategy

Specialization of Devki was accomplished through careful training and fine tuning of the base model in a multi-stage process with the goal of filling it with high level domain-specific knowledge.

The fine-tuning data set is a newly developed composite data set which is specially designed for the purpose of this research. It's made up of 500,000 instances across a variety of administrative tasks. Some of this data is proprietary and anonymized enterprise data, such as IT service management tickets, procurement requests and software license reports, that make up roughly 60% of the dataset. The other 40% is of high quality, artificially produced with a larger “teacher” model (GPT-4). This synthetic data, which was based on the “textbook-quality” data that has been used to successfully train SLMs such as the Phi series of Microsoft's “Textbook-Style” Knowledge Graphs (KGs), was essential for both training Devki with complex reasoning paflerns and to ensure that a broad range of administrative scenarios were covered and data was consistent and of quality.

The teacher model was then asked to not only give the correct output for a given task but also to give a step-by-step explanation of what they were doing, called a 'chain-of-thought' (CoT). This explanation of the reasoning steps was then incorporated into the training of Devki, the 'student' model, to predict both the final answer and the step-by-step reasoning. This learning from rich explanation traces technique is found to be very effective to improve the thinking skills of smaller models, so that they can get complex problem solving skills from their larger models.

The fine-tuning process was done by the Parameter-Efficient Fine-Tuning (PEFT) method of Low-Rank Adaptation (LoRA) using low-rank adapter matrices of rank 16 that were injected into the aflention layers of the model, with the majority of the pre-trained weights frozen.

This enabled the entire fine tuning of the model Devki from the specialised data set in This is indicative of the economic potential of the SLM specialization process, with the benefit of using a single server and four NVIDIA A100 80GB GPUs, which is significantly less time and expense than it would take to do full-model fine-tuning.

C. Perfiormance Benchmarks and Evaluation

A careful examination of Devki's performance was done using a high-quality baseline: a high-quality, generic proprietary large language model that is available via its public API, or "Baseline LLM. The assessment was based on three aspects of administration which are central to the functioning of an enterprise.

i. **Document Summarization:** The task was to summarise excerpts from internal financial reporting that were 1,000 words in length to a list of five points showing key information. This task requires the model to understand the domain-specific and dense texts and extract the main idea of the texts.

ii. **Information Extraction:** The objective was to extract information from a collection of 500 unstructured PDF invoices, specifically Vendor Name, Invoice ID, Due Date and Total Amount. This is a crucial step in accounts payable automation and it will evaluate how the model will detect and categorize certain entities in potentially noisy text.

iii. **Email Routing and Triage:** Email Routing and Triage: Classify a set of 1,000 emails originating from within the organisation into one of four classes (IT Support, HR Inquiry, Procurement Request, Spam) and provide the main purpose of the email. This is like a typical helpdesk automation process.

A set of task-specific and cross task metrics were used to measure performance. Summarization: ROUGE scores were used. Standard F1-Score and the Accuracy were used for information extraction and classification of emails. Table II shows the outcome of this comparative study, where we measured the Inference Latency (in millisec. per generated token) for both Devki and public API pricing across all the tasks, and calculated the estimated Cost per 1000 tasks based on the amortization of hardware costs for Devki and the public API pricing for the Baseline LLM.

Task	Metric	Devki (SLM)	Baseline (LLM)	% Improvement
Document Summarization	ROUGE-L	0.48	0.45	+6.7%
	Latency (ms/token)	25	150	+500%
	Cost ($/1k tasks)	$0.50	$4.00	+700%

Information Extraction	F1-Score	0.94	0.88	+6.8%
	Latency (ms/token)	18	120	+567%
	Cost ($/1k tasks)	$0.20	$2.50	+1150%
Email Routing & Triage	Accuracy	0.98	0.95	+3.2%
	Latency (ms/token)	15	110	+633%
	Cost ($/1k tasks)	$0.10	$1.80	+1700%

Table II: Performance benchmarks of the specialized SLM, Devki, versus a general-purpose Baseline LLM on three representative administrative tasks. The results demonstrate Devki's superior performance in both task-specific accuracy and efficiency metrics (latency and cost).

The empirical results do support the main argument of this paper. In each task, the dedicated SLM, Devki, not only matched, but surpassed the accuracy of the much larger Baseline LLM. A significant increase in the performance of the models for information extraction (+6.8%) and summarization (+6.7%) in F1-Score and ROUGE-L indicates the models' tangible value of domain-specific fine-tuning. Most impressively, the efficiency improvements are by a factor of 10. Devki showed 5-7x reduction in latency and 7-18 x reduction in operational cost and proved that the SLM paradigm is much more efficient and economically viable to provide enterprise automation solution.

IV. USHERING IN THE MENU-LESS ERA: SLMS AS THE NEW ENTERPRISE INTERFACE

Devki's success is not just a technical one; it's the beginning of a rethinking of enterprise user experience. With its low latency, high accuracy and on-premise security, specialized SLMs are the perfect engine to support the next generation of human-computer interaction, beyond the static and rigid GUI towards the fluid and intuitive paradigm of the Conversational User Interface.

A. From GUIs to Conversational Workflows: The Final Abstraction

Enterprise software interactions can be thought of as a stepping stone on the ascending path of abstraction. The Command-Line Interface (CLI) demanded users to learn a specialized language that involved a lot of syntax. The GUI abstracted this away and software began using visual elements such as icons, windows and menus, making it

accessible to a much wider range of people, but the GUI still requires its users to learn complex menu hierarchies and workflows designed by the software developer, sometimes traversing dozens of screens to complete a single complex task.

The last and greatest abstraction is the Conversational User Interface (CUI), which eliminates the interface altogether as all communication is done with human language, and it is not new; however there is not much use of these within the secure environment of a complex enterprise. Cloud-based generalist LLMs also suffer from high latency, data privacy concerns, and are not appropriate to orchestrate real time mission critical business processes.

Specialized SLMs such as Devki are able to overcome this impasse. The important thing about them is that they can run efficiently with low latency on premise without having to be cloud based, allowing the enterprise to experience the full power of the CUI. An SLM can be the 'Central Nervous System' as it gained this nickname, which intelligently understands what a user wants in their natural language and interacts with the business applications that are running on the user's behalf (such as ERPs, CRMs, ITSM platforms). It marks a “menu-less, click-less” period, in which the user's attention will be directed to what he or she wants to do instead of how to do it.

B. A Practical Application: A Multi-Step Workflow with Devki

To highlight this changing dynamic, let's take an example that is common, but often complex: Software License Management (SLM) audit. This process would be a time-consuming manual challenge in a traditional GUI-based environment, requiring an IT administrator to log into several databases, software deployment tools, procurement portals and ticketing systems and thoroughly cross reference data on screens and interfaces.

All this is streamlined as a single conversation with Devki:

Scenario: An IT administrator needs to audit the usage of Adobe Creative Cloud licenses in the marketing department.

- **Step 1 (Inventory and Reconciliation):**
  - **User:** *"Devki, get a report of all Adobe Creative Cloud software that is installed on the marketing department's laptops and compare this to the number of Adobe Creative Cloud*

licenses that we have in our asset management database.

- *Behind the Scenes: Devki interprets this command, identifies the entities ("Adobe Creative Cloud" and "marketing department") and infers its meaning ("inventory and compare"). Then it creates and sends API calls to the company's device management system and asset management database, fetches the corresponding data and then does the reconciliation. Analyze the problem(s) and make recommendations for improving the situation.*

- **Step 2 (Analysis and Recommendation):**
  - **Devki:** *I have identified 150 active installations, but only 120 of them purchased license, which means there are 30 licenses lacking. Based on my analysis, 15 of the unlicensed installations have not been used for more than 90 days. Do you want me to list this inactive user that I can resell for license?*
  - Devki has not only given the raw data, but also has used a business logic to focus on a particular problem (compliance gap) and a potential solution (reclaim the inactive licenses), showing its ability to reason in terms of tasks.
- **Step 3 (Action - Summarization and Communication):**
  - **User:** Yes, for all 15 active users who are not licensed, summarize the compliance risk in an email drak to the head of procurement and put in the cost of the true-up that we are expecting based on our enterprise agreement.
  - *Behind the Scenes:* Devki uses a knowledge base to retrieve the enterprise agreement document and to perform a summarization task to clearly define the risk, and generates a formafled email draft, which the administrator can then review and send.
- **Step 4 (Action - System Integration and Execution):**
  - **User:** " Decommission their inactive installations, and retrieve the licenses from the 15 users on ServiceNow, through the TI ticket."
  - Behind the Scenes: Devki authenticates with SOW via the ServiceNow API and creates total of 15 separate service tickets via programmatic, filling each with the relevant user and asset information, thus kicking off the final phase of the workflow.

The entire scenario here is an IT administrator that has finished a complicated audit across multiple systems, and is taking corrective actions—entirely from with a single conversational interface. They didn't need to know that certain systems contained the

data, how to generate a report on each and how to generate a service ticket. A natural and effective conversation guided the whole process, making it goal oriented. This is the true meaning of "menu-less, click-less," the paradigm of unprecedented efficiency and usability, enabled by specialized and enterprise grade SLMs.

C. CHALLENGES AND FUTURE DIRECTIONS

The arguments advanced in favour of the SLM paradigm are well founded, however, the full academic analysis will need to acknowledge some of the current constraints and move forward with a vision for the future evolution of SLM. There are of course, challenges to widespread adoption but also avenues for future research and innovation that are revealed by these challenges.

A. Overcoming the Limitations OFI Specialization

The specialisation is the key advantage of SLMs and the major constraint. The purpose of a model such as Devki – designed for administrative processes – is that it will be limited in scope of knowledge. It would have difficulty in responding An SLM's ability to answer "out-of-domain" queries (ones not in its training set) can suffer from poor results, with either incorrect or irrelevant replies.

Moreover, the problem of inherent bias won't go away just because they switch to smaller size. Specialised SLMs can be more easily identified and corrected for some biases, when compared to training on the entire internet, but have the potential to amplify biases found within a particular corporate or industry database. Rigorous auditing, fairness testing, and ongoing monitoring are required to ensure that specialised SLMs do not disproportionately reinforce historical inequities in their training data.

After all, there's an engineering and maintenance cost issue. The concept of a completely automated business may need not just one, but a dozen or more particular SLMs, one for finance, one for HR, one for legal and so on. The combination of developing, deploying and maintaining this wide range of models may be more of a heavy lift for an enterprise than the comparatively low-tech and low-effort call to a single central LLM API.

B. The Future ofi Composable, Agentic Enterprise AI

These challenges are solvable, however, not by going back to monolithic models, but

again, by moving towards a new, natural, logical and modular evolution of the SLM paradigm: an ecosystem of intelligent agents that are composable and modular. Rather than being just one grand "oracle" of AI, the future of enterprise AI is about a collection of highly specialized SLMs, each of which is expert in its own field, that work together. This "Lego-like" assembly of intelligence – scaling out rather than scaling up a single model – provides a way for a system to be more flexible, resilient and scalable than any one model.

This architectural vision tackles the restrictions of each individual SLM directly. An Intelligent Routing Layer is introduced to overcome the narrow scope. The first model is a lightweight and extremely efficient "dispatcher" which receives a user's natural language query. The job of this dispatcher would be to simply identify the intent behind this user's question and direct the question down to the appropriate specialist downstream, and not to provide the answer the users is seeking. A question such as: What do you expect to make in Q3?

The ‘Devki-Finance’ SLM would take ‘Make payment’ while ‘Reset my VPN password’ would be taken to the ‘Devki-IT’ SLM. This composable approach maintains the same performance, accuracy and security advantages of every specialist, but also offers a system capable of performing a wide variety of enterprise applications. It offers a scalable and manageable enterprise-wide solution, reducing the maintenance burden of a completely independent solution.

This vision of a composable, agentic system illuminates several critical **future research directions**:

i. **Efficient Model Routing:** The performance of the whole eco-system depends on the performance of the initial routing model, which needs to be efficient. New ultra-low latency intent recognition and dynamic dispatch models/algorithms for a heterogeneous, multi-model environment need to be developed.

ii. **Automated SLM Specialization:** There is a need to create new specialist SLM quickly and automatically with minimal engineering efforts. This might include the ability to create higher-quality techniques for few-shot or zero-shot domain adaptation by a common, pre-trained base SLM, so that enterprises can deploy new expert models on-demand as their business needs change.

iii. **Cross-Domain Reasoning and Collaboration:** The most complex enterprise workflows often span multiple administrative domains (e.g., a hiring process

involves HR, IT, and finance). Future work must explore frameworks and protocols that enable multiple specialized SLMs to collaborate, share context, and reason together to solve these complex, cross-functional problems, paving the way for truly autonomous enterprise agents.

CONCLUSION

Large Language Models have dominated the artificial intelligence discourse in the enterprise, with their projected size, as well as their generalist abilities. This paper, however, has suggested that in practice, and in the context of workflow automation in administration, the future isn't so big, but so smart and specialized. This shift in paradigm is fueled by the unassailable and interrelated benefits of being efficient, cost-effective, data secure, and accurate in a specific niche.

This work has been a good contribution to this new and developing area. First, it has offered a complete theory that grapples with the debate, showing the need to consider the SLM approach as more than just a performance trade-off, and instead as a strategy regarding TCO, security, and deployment feasibility when deciding on an enterprise use case. Second, it has developed a new task oriented SLM, called Devki, which provides a clear method for designing, training and implanting such SLM. The results of the empirical assessment of Devki against a state-of-the-art general-purpose LLM give empirical and quantitative evidence of the SLM paradigm's effectiveness in the representative administrative tasks. Last but not least, this paper has shown how these technically and economically better models can be thought of as the underlying technology for a completely different "menu-less, click-less" user experience, revolutionizing human-computer interaction in the workplace.

While specialization can be a challenge, it is also a glimpse of a more advanced and capable future – a system of self-governing, working together SLMs. Devki, and all the ideas it represents, gives a roadmap that is clear and actionable for this new generation of enterprise technology, which is smarter, more efficient, more secure, and more human. The quest for AI general intelligence will go on but the seemingly immediate and concrete shift to enterprise productivity will be a result of the concentrated power of the specialist.

REFERENCES

[25] Miraghaei, P., Moreschini, S., Kolehmainen, A., & Hästbacka, D. (2025). *Towards a Small Language Model Lifecycle Framework*. arXiv:2506.07695.

[19] Subramanian, S., Elango, V., & Gungor, M. (2025). *Small Language Models (SLMs) Can Still Pack a Punch: A survey*. arXiv:2501.05465.

[48] Nguyen, C. V., et al. (2024). *A Survey of Small Language Models*. arXiv:2410.20011.

[9] Splunk. (2025). *Language Models: SLM vs. LLM*. Splunk Blog.

[11] WEKA. (2025). *SLM vs. LLM: A Head-to-Head Comparison*. WEKA.io.

[17] Masood, A. (2024). *Multimodal Phi-4: How Small Language Models Are Quietly Reshaping Our World*. Medium.

[33] Number Analytics. (n.d.). *Mastering Task-Oriented NLP*. Number Analytics Blog.

[51] DFKI. (n.d.). *E&E: Efficient and explainable NLP models*. German Research Center for Artificial Intelligence.

[32] Meegle. (n.d.). *Natural Language Processing for Sustainability*. Meegle Topics.

[34] Wen, Y., Shi, F., & Mou, L. (2025). *Knowledge Distillation for Language Models*. Proceedings of the 2025 Annual Conference of the NAACL: HLT (Tutorial Abstracts).

[36] Wen, Y., Shi, F., & Mou, L. (2025). *Knowledge Distillation for Language Models*. ACL Anthology.

[35] Anonymous. (2025). *Distilling Data and Rewards for Language Model Distillation*. arXiv:2502.19557.

[29] Red Hat. (n.d.). *What is parameter efficient fine-tuning (PEFT)?*. Red Hat Topics.

[30] Kanerika. (n.d.). *Parameter-efficient Fine-tuning (PEFT): The Future of NLP*. Kanerika Blogs.

[31] Sachin, T. (n.d.). *Parameter-Efficient Fine-Tuning for Models: Categories and Algorithms*. Medium.

[10] Red Hat. (2025). *LLMs vs. SLMs: What's the difference?*. Red Hat Topics.

[12] Arcee.ai. (2025). *7 Key Advantages of SLM Over LLM for Businesses*. Arcee.ai Blog.

[20] Lamatic.ai. (2025). *Complete SLM vs LLM Guide for Faster, Cost-Effective AI Solutions*. Lamatic.ai Blog.

[14] HorizonIQ. (2025). *SLM vs LLM: The Ultimate Guide to Choosing the Right AI Model*. HorizonIQ Blog.

[13] PremAI. (2025). *Fine-Tuning Small Language Models: A Practical Guide*. PremAI Blog.

[28] Anonymous. (2025). *Fine-Tuning SLMs vs. Prompting LLMs for Structured Output Generation*. arXiv:2505.24189.

[1] InvGate. (n.d.). *Sokware License Management (SLM): A Complete Guide*. InvGate ITAM.

[2] ServiceNow. (n.d.). *Service Level Management*. ServiceNow Products.

[37] Dataloop.ai. (n.d.). *Model Library: llmware_slim-summary*. Dataloop.ai.

[38] Analytics Vidhya. (2023). *Build A Text Summariser Using LLMs with Hugging Face*. Analytics Vidhya Blog.

[39] Purplescape. (2024). *Understanding Small Language Models (SLMs) and Its Applications*. Purplescape.

[42] Google. (n.d.). *Add a routing seWing*. Google Workspace Admin Help.

[25] Miraghaei, P., Moreschini, S., Kolehmainen, A., & Hästbacka, D. (2025). *Towards a Small Language Model Lifecycle Framework*. arXiv:2506.07695. [52] Subramanian, S., Elango, V., & Gungor, M. (2025).
Small Language Models (SLMs) Can Still Pack a Punch: A survey. arXiv:2501.05465.

[50] Anonymous. (2025). *The Future of Agentic AI is Small*. arXiv:2506.02153.

[26] Aussie AI. (n.d.). *Small Language Models*. AussieAI.com.

[48] Nguyen, C. V., et al. (2024) *A Survey of Small Language Models*. arXiv:2410.20011.

[18] Microsoft Azure. (n.d.). *What are small language models?*. Azure Cloud Computing Dictionary.

[47] Shakudo. (n.d.). *SLMs vs. LLMs: Choosing the Right AI Solution for Your Business*. Shakudo Blog.

[15] Ajith P. (2025). *Small Language Models (SLM): The Enterprise AI Revolution*. Ajithp.com.

[46] Cufler Consortium. (n.d.). *Thinking Small to Win Big: The Rise of SLMs in Enterprise AI*. Cufler.com.

[16] Kamran, A. (n.d.). *SLMs (Small Language Models) and the 7 Key Scenarios Where They Surpass LLMs*. Medium.

[49] Reddit User. (2024). *Enterprise challenges in deploying open-source LLMs to production at scale*. r/LocalLLaMA.

[4] Mosaicx. (2025). *12 Ways Enterprise Companies Use Conversational AI*. Mosaicx Blog.

[40] TEKsystems. (2024). *Natural Language Processing (NLP) for Business Transformation*. TEKsystems Insights.

[7] McKinsey & Company. (2025). *Superagency in the workplace: Empowering people to unlock AI's full potential*. McKinsey Digital.

[8] IBM. (2025). *AI and the Future of Work*. IBM Think Insights.

[5] Tribe AI. (2025). *How 3 Companies Automated Manual Processes Using NLP*. Tribe AI Blog.

[6] Iauro. (2025). *Automating Routine Tasks with NLP: Boosting Operational Efficiency in Enterprises*. Iauro Blog.

[41] Provalet. (2025). *How NLP is Revolutionizing Communication and Document Processing Automation Today*. Provalet Guides.

[3] Akitra. (2024). *NLP in Automating Compliance Documentation & Reporting*. Akitra Blog.

[44] AIMultiple. (2025). *Conversational UI: 6 Best Practices in 2025*. AIMultiple Research.

[21] The Story. (2024). *What is a conversational user interface (CUI)?*. The Story Journal.

[45] XenonStack. (2025). *Guide to Conversational User Interface Best Practices and CUI Tools*. XenonStack Insights.

[22] ACI Infotech. (2025). *The Future of Enterprise Apps with Conversational Interfaces*. ACI Infotech Blogs.

[53] Zendesk. (2024). *What is a conversational interface?*. Zendesk Blog.

[23] UXtweak. (n.d.). *Natural language interface*. UXtweak UX Glossary.

[24] Wikipedia. (2025). *Natural-language user interface*.

[9] Splunk. (2025). *Language Models: SLM vs. LLM*. Splunk Blog.

[11] WEKA. (2025). *SLM vs. LLM: A Head-to-Head Comparison*. WEKA.io.

[10] Red Hat. (2025). *LLMs vs. SLMs: What's the difference?*. Red Hat Topics.

[12] Arcee.ai. (2025). *7 Key Advantages of SLM Over LLM for Businesses*. Arcee.ai Blog.

[14] HorizonIQ. (2025). *SLM vs LLM: The Ultimate Guide to Choosing the Right AI Model*. HorizonIQ Blog.

[13] PremAI. (2025). *Fine-Tuning Small Language Models: A Practical Guide*. PremAI Blog.

[28] Anonymous. (2025). *Fine-Tuning SLMs vs. Prompting LLMs for Structured Output Generation*. arXiv:2505.24189.

[27] Subramanian, S., Elango, V., & Gungor, M. (2025). *Small Language Models (SLMs) Can Still Pack a Punch: A survey*. arXiv:2501.05465.

[43] Xurrent. (2025). *Natural language processing (NLP): The bridge between human communication and intelligent automation*. Xurrent Blog.

APPENDICES

Appendix A: Prompt Templates

In the evaluation that took place in Section III. To make a fair and consistent comparison, the following prompt templates were used to query the Baseline LLM (C).

Task 1: Document Summarization

You are a highly qualified financial analyst. Please review the excerpt of the financial report below and give a brief bullet-point summary of the key findings. Emphasize on the revenue trend, cost drivers and overall profitability.

Task 2: Information Extraction

You are a computerised data entry operator. Given the following invoice document, break it down to get the following information in a JSON object. The mandatory fields are: "vendorName", "invoiceID", "dueDate" and "totalAmount". Make sure date is in the format YYYY-MM-DD and amount is in a float number.

Task 3: Email Routing and Triage

You are a computer-based help desk system. Put the following e-mail in one of these four categories: "IT Support", "HR Inquiry", "Procurement Request", or "Spam". Next, write a one sentence summary of the main objective of the user. Answer in JSON, include key "category" and key "intent".

Appendix B: Additional Results and Error Analysis

The Information Extraction task from Section III.C is broken down into more detail in this appendix. Although Devki has a higher overall F1-Score, there are important differences between the errors made by both models in a qualitative analysis of these errors.

- **Baseline LLM Errors:** Most of the errors (around 70%) by the Baseline LLM were "formafling hallucinations. The model was able to accurately recognize the entities but did not consistently follow the strict JSON output format required, and some of the output contained conversational text, and the data types were incorrect (e.g. returning a

total amount as a string including a currency symbol).

- **Devki Errors:** Most of the errors that Devki made were known as "entity confusion" errors (around 85%) that were often present in expense/invoice layouts with lots of data, where Devki sometimes confused the "shipping date" with the "due date". This implies that its domain-specific training was not only very robust to the output format, but more training on a wider range of document layouts will further enhance its entity recognition performance.

Works cited

1. What is Software License Management (SLM)? Process And Tools - InvGate, accessed July 30, 2025,https://invgate.com/itsm/it-asset-management/software-license-management
2. Service Level Management - ServiceNow, accessed July 30, 2025, https://www.servicenow.com/products/service-level-management.html
3. NLP in Automating Compliance Documentation & Reporting | Akitra, accessed July 30, 2025, https://akitra.com/nlp-in-automating-compliance-documentation-reporting/
4. 12 Ways Enterprise Companies Use Conversational AI - Mosaicx, accessed July 30, 2025, https://www.mosaicx.com/blog/enterprise-conversational-ai-uses
5. How 3 Companies Automated Manual Processes Using NLP | Tribe AI, accessed July 30, 2025, https://www.tribe.ai/applied-ai/how-3-companies-automated-manual-processes-using-nlp
6. Automating Routine Tasks with NLP: Boosting Operational Efficiency ..., accessed July 30, 2025, https://iauro.com/automating-routine-tasks-with-nlp-boosting-operational-effici ency-in-enterprises/
7. AI in the workplace: A report for 2025 | McKinsey, accessed July 30, 2025, https://www.mckinsey.com/capabilities/mckinsey-digital/our-insights/superagenc y-in-the-workplace-empowering-people-to-unlock-ais-full-potential-at-work
8. AI and the Future of Work | IBM, accessed July 30, 2025, https://www.ibm.com/think/insights/ai-and-the-future-of-work
9. LLMs vs. SLMs: The Differences in Large & Small Language Models ..., accessed July 30, 2025, https://www.splunk.com/en_us/blog/learn/language-models-slm-vs-llm.html
10. SLMs vs LLMs: What are small language models? - Red Hat, accessed July 30, 2025, https://www.redhat.com/en/topics/ai/llm-vs-slm
11. SLM vs LLM: The Key Differences - WEKA, accessed July 30, 2025, https://www.weka.io/learn/ai-ml/slm-vs-llm/
12. 7 Key Advantages of SLM Over LLM for Businesses - Arcee AI, accessed July 30, 2025, https://www.arcee.ai/blog/7-key-advantages-of-slm-over-llm-for-businesses
13. Fine-Tuning & Small Language Models - Prem AI Blog, accessed July 30, 2025, https://blog.premai.io/fine-tuning-small-language-models/

14. SLM vs LLM: Key Differences and Use Cases | HorizonIQ, accessed July 30, 2025, https://www.horizoniq.com/blog/slm-vs-llm/
15. Small Language Models: The $5.45 Billion Revolution Reshaping Enterprise AI, accessed July 30, 2025, https://ajithp.com/2025/05/26/small-language-models-slm/
16. SLMs (Small Language Models) and the 7 key scenarios where they surpass LLMs | by Arman Kamran | Medium, accessed July 30, 2025, https://medium.com/@armankamran/slms-small-language-models-and-the-7-key-scenarios-where-they-surpass-llms-b548d73de85e
17. Multimodal Phi 4- How Small Language Models Are Quietly Reshaping Our World - Medium, accessed July 30, 2025, https://medium.com/@adnanmasood/multimodal-phi-4-how-small-language-models-are-quietly-reshaping-our-world-c3251286f241
18. What Are Small Language Models (SLMs)? | Microsoft Azure, accessed July 30, 2025, https://azure.microsoft.com/en-us/resources/cloud-computing-dictionary/what-are-small-language-models
19. Small Language Models (SLMs) Can Still Pack a Punch: A survey - arXiv, accessed July 30, 2025, https://arxiv.org/html/2501.05465v1
20. Complete SLM vs LLM Guide for Faster, Cost-Effective AI Solutions, accessed July 30, 2025, https://blog.lamatic.ai/guides/slm-vs-llm/
21. Conversational interface. What is it, and what are its uses? - The Story, accessed July 30, 2025, https://thestory.is/en/journal/what-is-a-conversational-user-interface-cui/
22. The Future of Enterprise Apps: AI-Powered Conversational ..., accessed July 30, 2025, https://www.aciinfotech.com/blogs/future-of-enterprise-apps-with-conversational-interfaces
23. Natural language interface | UXtweak, accessed July 30, 2025, https://www.uxtweak.com/ux-glossary/natural-language-interface/
24. Natural-language user interface - Wikipedia, accessed July 30, 2025, https://en.wikipedia.org/wiki/Natural-language_user_interface
25. Towards a Small Language Model Lifecycle Framework - arXiv, accessed July 30, 2025, https://arxiv.org/html/2506.07695v1
26. Small Language Models - Aussie AI, accessed July 30, 2025, https://www.aussieai.com/research/small-models
27. Small Language Models (SLMs) Can Still Pack a Punch: A survey, accessed July 30, 2025, https://arxiv.org/abs/2501.05465
28. Fine-Tune an SLM or Prompt an LLM? The Case of ... - arXiv, accessed July 30, 2025, https://arxiv.org/abs/2505.24189
29. What is parameter-efficient fine-tuning (PEFT)? - Red Hat, accessed July 30, 2025, https://www.redhat.com/en/topics/ai/what-is-peft
30. Parameter-Efficient Fine-Tuning (PEFT): Optimizing LLMs - Kanerika, accessed July 30, 2025, https://kanerika.com/blogs/parameter-efficient-fine-tuning/
31. Parameter-Efficient Fine-Tuning for Models: Categories and Algorithms - Medium,

accessed July 30, 2025, https://medium.com/@techsachin/parameter-efficient-fine-tuning-for-models-c ategories-and-algorithms-4481fb2bdef0
32. Natural Language Processing For Sustainability - Meegle, accessed July 30, 2025, https://www.meegle.com/en_us/topics/natural-language-processing/natural-lang uage-processing-for-sustainability
33. Mastering Task-Oriented NLP - Number Analytics, accessed July 30, 2025, https://www.numberanalytics.com/blog/mastering-task-oriented-nlp
34. Knowledge Distillation for Language Models - ACL Anthology, accessed July 30, 2025, https://aclanthology.org/2025.naacl-tutorial.4.pdf
35. Distill Not Only Data but Also Rewards: Can Smaller Language Models Surpass Larger Ones? - arXiv, accessed July 30, 2025, https://arxiv.org/html/2502.19557v1
36. Knowledge Distillation for Language Models - ACL Anthology, accessed July 30, 2025, https://aclanthology.org/2025.naacl-tutorial.4/
37. Slim Summary · Models · Dataloop, accessed July 30, 2025, https://dataloop.ai/library/model/llmware_slim-summary/
38. Text Summariser Using LLMs with Hugging Face - Analytics Vidhya, accessed July 30, 2025, https://www.analyticsvidhya.com/blog/2023/07/build-a-text-summariser-using-ll ms-with-hugging-face/
39. Understanding Small Language Models (SLMs) and Its Applications, accessed July 30, 2025, https://purplescape.com/understanding-small-language-models-slms-and-its-ap plications/
40. Natural Language Processing (NLP) for Business Transformation ..., accessed July 30, 2025, https://www.teksystems.com/en/insights/version-next-now/2024/natural-languag e-processing
41. How NLP is Revolutionizing Communication and Document ..., accessed July 30, 2025, https://www.provalet.io/guides-posts/the-role-of-nlp-in-automating-communicat ion-and-document-processing
42. Add Gmail Routing settings - Google Workspace Admin Help, accessed July 30, 2025, https://support.google.com/a/answer/6297084?hl=en
43. Natural language processing (NLP): The bridge between human ..., accessed July 30, 2025, https://www.xurrent.com/blog/nlp-transforms-itsm-operations
44. Conversational UI: 6 Best Practices in 2025 - Research AIMultiple, accessed July 30, 2025, https://research.aimultiple.com/conversational-ui/
45. Guide to Conversational User Interface Best Practices and CUI Tools, accessed July 30, 2025, https://www.xenonstack.com/insights/conversational-user-interface-best-practic es
46. Thinking Small, Winning Big: The Rise of SLMs in Enterprise AI | Cutter Consortium, accessed July 30, 2025, https://www.cutter.com/article/thinking-small-winning-big-rise-slms-enterprise-ai
47. SLMs vs LLMs: Choosing the Right Enterprise AI Solution for Your Business | Shakudo, accessed July 30, 2025, https://www.shakudo.io/blog/slms-vs-llms-choosing-the-right-ai-solution-for-yo ur-business
48. A Survey of Small Language Models - arXiv, accessed July 30, 2025,

https://arxiv.org/html/2410.20011v1
49. Enterprise Challenges in Deploying Open-Source LLMs at Scale: Where Do You Struggle Most? : r/LocalLLaMA - Reddit, accessed July 30, 2025, https://www.reddit.com/r/LocalLLaMA/comments/1h0bh2r/enterprise_challenges_in_deploying_opensource/
50. Small Language Models are the Future of Agentic AI - arXiv, accessed July 30, 2025, https://arxiv.org/html/2506.02153v1
51. E&E Group: Efficient and explainable NLP models - Deutsches Forschungszentrum für Künstliche Intelligenz, accessed July 30, 2025, https://www.dfki.de/en/web/research/research-departments/multilinguality-and-language-technology/ee-team
52. Small Language Models (SLMs) Can Still Pack a Punch: A survey - ResearchGate, accessed July 30, 2025, https://www.researchgate.net/publication/387953927_Small_Language_Models_SLMs_Can_Still_Pack_a_Punch_A_survey
53. What is a conversational interface? - Zendesk, accessed July 30, 2025, https://www.zendesk.com/blog/conversational-interface/