Pro-QuEST: A Prompt-chain based Quiz Engine for testing Specialized Technical Product Knowledge

Sujatha Das Gollapalli,[1] **Mouad Hakam,**[1] **Mingzhe Du,**[1,2]
See-Kiong Ng,[1] **Mohammed Hamzeh**[3]
[1]Institute of Data Science, National University of Singapore
[2]College of Computing and Data Science, Nanyang Technological University
[3]Cisco Systems, Inc. Austin, TX, U.S.A.
{idssdg,mouad.hk,mingzhe,seekiong}@nus.edu.sg, mhamzeh@cisco.com

Abstract

As large language models (LLMs) rapidly evolve and proliferate, technology companies such as *Cisco* face the difficult challenge of selecting the most suitable model for downstream tasks that demand deep, domain-specific product knowledge. Specialized benchmarks can not only inform this decision making but also be leveraged as quizzes to effectively train engineering and marketing personnel on novel product offerings in a continually growing *Cisco* product space.

We present Pro-QuEST, our **Pro**mpt-chain based **Qu**iz **E**ngine using state-of-the-art LLMs for generating multiple-choice questions on **S**pecialized **T**echnical products. In Pro-QuEST, we first identify key terms and topics from a given professional certification textbook or product guide, and generate a series of multiple-choice questions using domain-knowledge guided prompts. We show LLM benchmarking results with the question benchmarks generated by Pro-QuEST using a range of latest open-source, and proprietary LLMs and compare them with expert-crafted exams and review questions to derive insights on their composition and difficulty. Our experiments indicate that though there is room for improvement in Pro-QuEST to generate questions of the complexity levels seen in expert-designed certification exams, question-type based prompts provide a promising direction to address this limitation. In sample user studies with *Cisco* personnel, Pro-QuEST was received with high optimism for its practical usefulness in quickly compiling quizzes for self-assessment on knowledge of novel products in the rapidly changing tech sector.

1 Motivation

Large Language Models (LLMs) have emerged as a transformative technology for various tasks resulting in their current wide-adoption across several technology industries (Raza et al., 2025; Palen-Michel et al., 2024; Company, 2023). Though LLMs demonstrate excellence at tasks requiring general language understanding such as text analysis, content generation, and summarization, their capabilities and limits for knowledge-intensive domains such as finance, engineering, cybersecurity, and healthcare is still a subject of active research (Fei et al., 2024; Xie et al., 2024; Ouyang et al., 2024). In particular, though Retrieval Augmented Generation (RAG) and knowledge integration (Song et al., 2025; Lewis et al., 2020) have helped in addressing limitations such as hallucination and content grounding, state-of-the-art LLMs still fall behind on tasks requiring complex, novel or multi-step reasoning, where tacit or proprietary knowledge is required, and where contexts and prior experience inform decision making (Chen et al., 2024; Yang et al., 2025; Xu et al., 2025; Kim et al., 2025).

Concurrent with the above research, newer LLMs are being released frequently each with unique architectures, and capabilities, and fine-tuned for specific capabilities (Xiao et al., 2025; Rizzatti, 2025; Wang et al., 2025a). In this changing landscape of LLMs and ongoing research on the promise and limitations of LLMs for specific domains, industry players have to make model choices under economic constraints (Howell et al., 2023). Against this context, standardized benchmarks which quantify LLMs' capabilities via precise performance metrics, characterize knowledge contamination, and provide guidance on making informed choices comprise **crucial** assets for a company. Indeed, both LLM benchmarking and benchmark generation now form core topics of active investigation in several domains (Fei et al., 2024; Ouyang et al., 2024; Xie et al., 2024).

In this study, we investigate the creation of domain-specific LLM benchmarks for *Cisco*, a technology company providing thousands of net-

Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics
Volume 3: System Demonstrations, pages 46–60

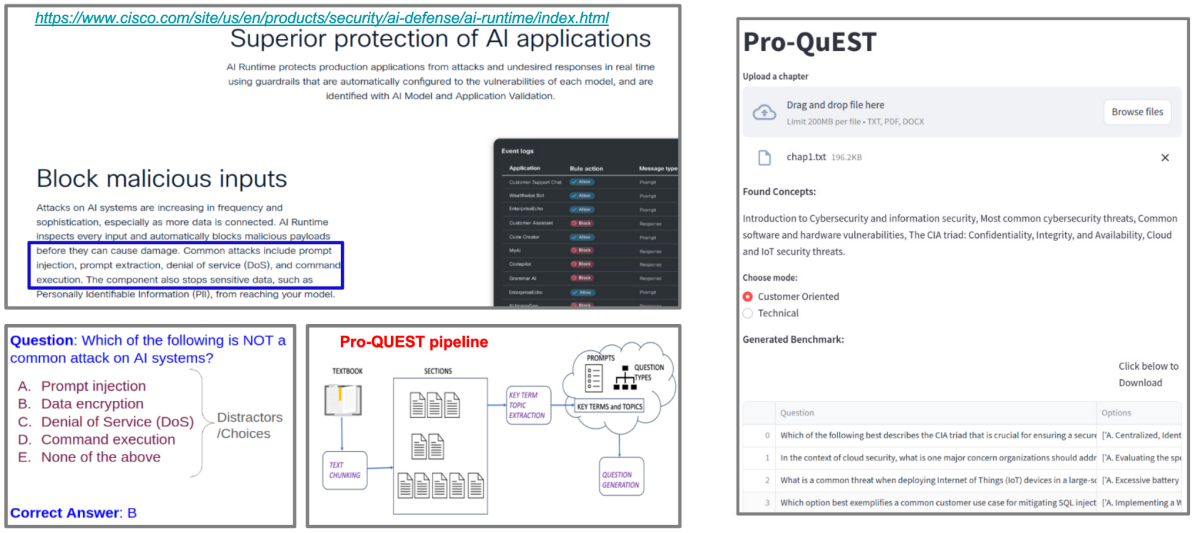

Figure 1: Our main task is illustrated with a sample input section and an LLM-generated multiple-choice question. The modules comprising the Pro-QuEST-pipeline and a screenshot of our web-based demo are also shown.

working products and services to customers across the world. *Cisco* contends with other competitors such as *Arista Networks*, *Dell Technologies*, and *Huawei* for market share pertaining to various device and software offerings in networking and cybersecurity, a rapidly developing sector. Therefore, not only is it critical for company marketing and sales professionals to be well-aware of the various features of devices from *Cisco* and competition, customer support engineers need to be familiar with intricate operational features to provide effective troubleshooting for clients. At the same time, as newer LLMs are made available, a quantitative assessment of their capabilities on internal tasks such as service request handling, question answering, and predictive analytics becomes inevitable (Yang et al., 2023; Tanhaei et al., 2024; Gollapalli et al., 2025; Saini et al., 2025).

How can we create representative benchmarks that measure specialized domain knowledge for guiding LLM selection for internal tasks and for training personnel on the complex and rapidly growing product mix in Cisco? We developed Pro-QuEST, our **Pro**mpt-chain based **E**ngine for generating benchmark **Qu**izzes on the **S**pecialized, **T**echnical knowledge in *Cisco* to address this question. In this demo paper, we describe the core components of Pro-QuEST and our experiments with generated benchmarks on content from *Cisco* textbooks used by technical personnel preparing for entry-level and advanced-level certifications (CCNA 200-301[3] and CCNP 350-701[4]). We highlight Pro-QuEST's exciting potential for training sales and marketing personnel on the novel product offerings with respect to their facts and features and provide insightful experiments on improving automatic benchmark creation for highly specialized, technical domains with SOTA LLMs. A **web-based demo** of Pro-QuEST was showcased at a recent Cisco Live event[1] and is available at https://nlp-demos.online/qg/ with an illustrative **video** available at https://github.com/mouad157/Cisco-benchmark/blob/main/EACL_ProQuEST.mp4.

2 System Components

Question answering (QA) datasets are commonly used in LLM benchmarking studies for specialized domains since QA accuracy provides a quantifiable measure of "knowledge" of a specific domain (Fei et al., 2024; Ouyang et al., 2024; Xie et al., 2024; Chen et al., 2025). Following our objective to create QA datasets for *Cisco*, we follow recent works (Choi et al., 2025; Xiong et al., 2024; Camarata et al., 2025; Dalvi et al., 2024) and apply document grounded multiple-choice question generation using SOTA LLMs in Pro-QuEST. An anecdotal illustration of our main task and our processing pipeline can be found in Figure 1 along with a screenshot of our web-based interface.

Pro-QuEST uses **prompt chaining** (Wu et al., 2022; Sun et al., 2024), a widely used technique in LLMs to break down complex tasks into a series of simpler tasks by using the output of one

[1] https://www.ciscolive.com/apjc.html

prompt as the input to the next prompt in the chain.[2] Prompt chains reduce the "cognitive load" for an LLM through explicit instructions on the steps involved in solving complex tasks and were shown to improve output quality and reduce hallucination through context retention between prompts. Considering context limitations in LLMs and the often lengthy nature of input documents (such as textbooks) that represent "knowledge", we accomplish three tasks in Pro-QuEST through a prompt chain as follows:

1. Section Chunking: We use LLM prompts on the first few pages of a long *text* document to identify the document type (such as product guides, textbooks, research papers, configuration matrix documents in *Cisco*), as well as other metadata information. These prompts aim to extract content organization information in the input document (for example, "table of contents"). The identified section headings or chapter titles are used to split a lengthy input document into smaller cohesive text chunks for further processing.

2. Key Terms and Topics Identification: From the sections identified in the previous step, we identify and collate the topical keyphrases and overarching themes using LLM prompts. Extraction of topical keyphrases is a widely-studied topic in NLP due to their effectiveness in representing and summarizing vast amounts of information from lengthy documents (Boudin and Aizawa, 2025). Indeed, keyphrases are widely used for various retrieval, analytical, and organizational tasks as well as to ground question generation (Willis et al., 2019; Wang et al., 2020; Zhang and Zhu, 2021).

3. Multiple-Choice Question Generation: Finally, the content and keyphrases from the previous two steps are combined with a diverse list of LLM prompts to generate multiple-choice questions (MCQs), the answer options or *distractors* for the questions, and the lists of correct answers. MCQs are prevalent in Education as well as LLM benchmarking since they can be designed for various levels of learning complexity and allow for an efficient, quantitative assessment (Camarata et al., 2025; Jovanovska, 2018).

Our above pipeline ensures coverage of all main topics of a lengthy document. When LLM prompts fail to extract sections in Step-1 (for example, when a content listing is missing), we first identify "section headings" using a heuristic algorithm that couples stylistic cues along with section length thresholds. For example, most words in section header sentences are capitalized and their average length is smaller than a typical sentence in the main body.[7] Our zero-shot LLM prompt templates are included in Tables 5, 10, and 11 of the Appendix.

In ongoing research, techniques such as chain-of-thought reasoning (Sprague et al., 2025), and in-context learning (Dong et al., 2024) are being employed to generate complex questions in specific domains such as Finance and Medicine (Choi et al., 2025; Liang et al., 2023). Such **overt** question design knowledge from *Cisco* experts was not available to us. We therefore focus on MCQ generation with simple prompts and compare them with available expert-compiled questions to derive insights that can inform future prompt design. Model-generated questions, regardless of the complexity of prompts employed, need expert validation for specialized domains. While this human validation is in progress, in this paper, we provide quantitative evaluation by characterizing question answering performance and comparing generated benchmarks with the available expert-compiled questions for their composition and difficulty.

3 Experiments

Datasets: The datasets for developing and testing Pro-QuEST were provided by *Cisco* and include two textbooks that are official preparation guides for the certification exams: (1) Cisco Certified Network Associate, an **entry-level** certification covering foundational networking skills (CCNA 200-301[3]) and (2) Cisco Certified Network Professional, an **advanced** certification for professionals for operating core security technologies (CCNP 350-701[4]). Both textbooks are long and image-heavy documents containing 29 and 11 content chapters, respectively. In this study, we only focused on the textual content, and sampled four chapters from each textbook for experiments. We refer to the expert-designed review questions available with each chapter from these textbooks with the label "Book" in our experiments. The textbooks also contained expert-specified key terms that we used to evaluate Step-2 of our Pro-QuEST-pipeline (Section A.1).

[2] https://www.promptingguide.ai/techniques/prompt_chaining

[3] https://www.oreilly.com/library/view/ccna-200-301-official/9780136755562/

[4] https://www.oreilly.com/library/view/ccnp-and-ccie/9780138221287/

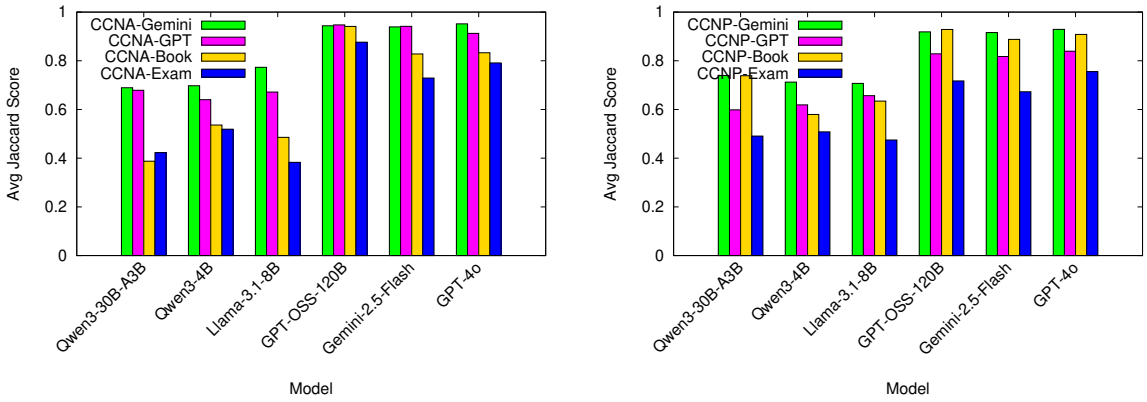

Figure 2: Results of LLM Benchmarking. The average Jaccard scores for CCNA and CCNP datasets are shown.

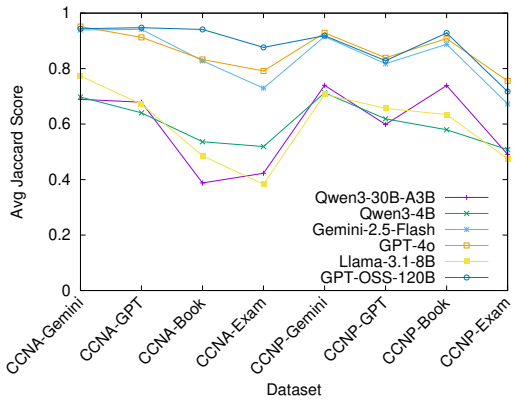

Figure 3: Model QA performance on our datasets.

Cisco also provided access to the question banks of the two certification exams. Unlike textbooks, this content is proprietary making it unlikely to be pre-trained knowledge for SOTA LLMs. Exam and textbook questions are created by domain experts and cover the spectrum of fundamental to complex topics ("CCNA versus CCNP") as well as difficulty range ("Book versus Exam"). A summary of our datasets is provided in Table 1. The LLM-generated questions using the sections from CCNA and CCNP textbooks are indicated using suffixes '-GPT' and '-Gemini'.

CCNA-Book	187	CCNP-Book	49
CCNA-Exam	400	CCNP-Exam	572
CCNA-GPT	165	CCNP-GPT	159
CCNA-Gemini	148	CCNP-Gemini	169

Table 1: Question datasets are shown with the number of questions in each set

We investigated keyphrase/keytopic extraction and question generation in Pro-QuEST using state-of-the-art LLMs–(1) GPT-4o from OpenAI,[5] and (2) Gemini-2.5-Flash.[6] Our choice of models was influenced by the available best performing, versatile models at the time of experiments, as well as pricing and context length considerations. For LLM benchmarking experiments, we selected small to large, open-source and proprietary models: *Qwen3-4B, Qwen3-30B-A3B*,*Llama-3.1-8B*, *GPT-OSS-120B, GPT-4o* and *Gemini-2.5-Flash*. We include their details in Table 8 of the Appendix. Our code and prompts are shared on GitHub[7] for research purposes with further details on dataset processing and experimental settings included in the Appendix.

3.1 LLM Benchmarking Results

Question Answering Performance: We evaluated a range of recent LLM models on all our datasets from Table 1 on the Question Answering (QA) task. QA performance was measured using Jaccard accuracy that measures the set overlap between predicted answers ('A') and the correct answers ('B') as $\frac{|A \cap B|}{A \cup B|}$.

As can be noticed in the performance plots of Figures 2 and 3, QA accuracies of the smaller models (from Qwen and Meta) are significantly lower than that of the much larger proprietary models as well as the 120B parameter model from OpenAI (*GPT-OSS-120B*). This is not surprising since larger LLMs which have several scales higher numbers of parameters can be expected to "know" more and demonstrate higher QA performance. We note that the QA performance is consistently, significantly higher for model generated questions (*-

[5] https://openai.com/api/
[6] https://aistudio.google.com/
[7] https://github.com/mouad157/Cisco-benchmark

GPT and *-Gemini datasets) compared to the Exam datasets suggesting that LLM-generated benchmark questions may be easier to answer than expert-designed benchmark questions.

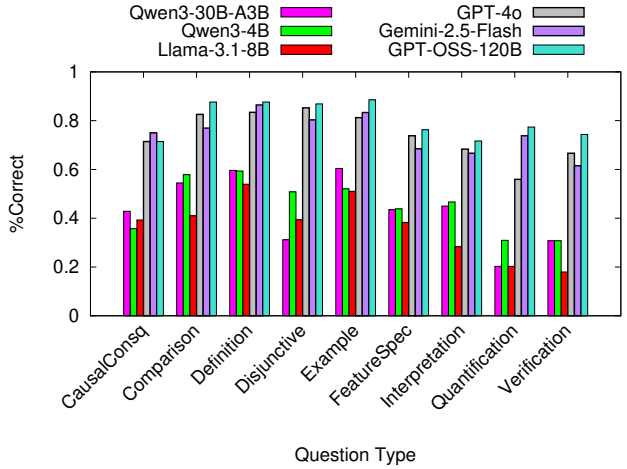

Figure 4: QA performance by question type

Question Difficulty: We grouped questions from our datasets into the following difficulty categories: ***easy***–answered correctly by all small models, ***medium***–answered incorrectly by all small models and correctly by majority of the large models, and ***hard***–answered incorrectly by all large models. In Table 2, the percentages of ***easy/medium/hard*** questions as per the above definitions are illustrated for all the datasets. Unlike Exams where the ***easy*** questions comprise 10-16% of the datasets, the LLM-generated benchmarks have a considerably higher 30-40% ***easy*** questions. Similarly, the questions in the ***hard*** category are considerably higher 9-18% in Exams compared to 1-12% in the LLM-generated benchmarks. The numbers of ***easy*** and ***hard*** questions in Books seem to represent averages of these other two sources.

Question Type Analysis: We further analyzed benchmark questions by employing a question-type taxonomy available from prior works (Zhao and Jiang, 2010; Nielsen et al., 2008). Question-types represent the nature of information sought in the answer to a given question. For instance, a 'Definition'-type question asks for how a given concept may be defined (Example: "*In a LAN, which of the following terms best equates to the term VLAN?*") whereas a 'Quantification' question asks about quantitative aspects of a situation (Example: "*What is the maximum number of distribution switches that can be deployed within a hierarchical LAN design building block?*"). A list of example multiple-choice questions from *Cisco* datasets for our twelve question types as well as details of question type prediction are included in Section A.3 of the Appendix.

We show the QA performance of various models grouped by question types in Figure 4. Similar to earlier experiments, smaller models are significantly worse than the larger LLMs across the question types. Indeed, performance with smaller models is particularly limited on *Interpretation/Quantification/Disjunctive* question types.[8] These categories are arguably challenging for the larger models as well since the QA performance on these types is lower compared to types such as *Definition/Comparison/Example*.

Overall the *GPT-OSS-120B* model, the latest, largest open-source offering from OpenAI which is also a reasoning model, closely outperforms GPT-4o and Gemini-2.5-Flash on all question types but one. We would like to highlight that for certain question types, such as *Quantification/Interpretation*, it is highly likely that chain-of-thought style complex prompts yield better results (Sprague et al., 2025). Moreover, companies such as OpenAI have multiple LLM offerings designed for specific use-cases. In this study, we consider LLMs designed for overall versatility and employ simple QA/QG prompts which can be employed across all LLMs uniformly (Table 6). We posit that this setting is more reflective of LLM's role as a "stand-in exam taker".

The question type spreads for CCNP datasets are shown in Figure 5 with those for CCNA included in the Appendix. The question type "*Feature Specification*" dominates the Exam benchmark and occurs only half as frequently in the GPT-generated benchmark, whereas the opposite is the case for the "*Definition*" question type. Given the significantly higher QA performance for this latter type, it is not surprising that all LLMs uniformly under-perform on the Exam questions in the benchmarking experiments (Figures 2 and 3).

We conducted experiments on incorporating specific question type into the LLM prompts (Tables 10 and 12 in the Appendix). While initial, anecdotal results with type-augmented prompts seem promising, this research and expert evalua-

[8]We observe here that in addition to the actual answer, in a considerable number of cases, smaller models do not follow prompt guidance with respect to output format and add explanations and reasoning process, despite explicit directions not to, resulting in errors during output parsing.

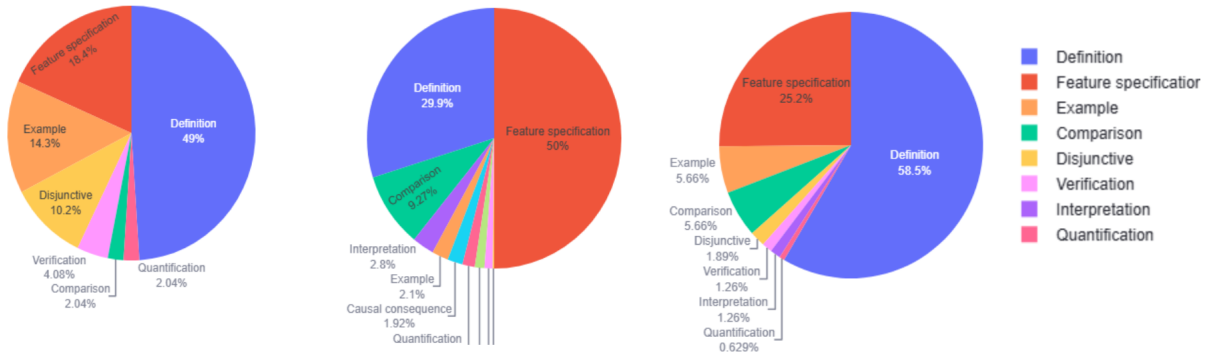

Figure 5: Question Type Distribution for CCNP-Book (left), CCNP-Exam (center), and CCNP-GPT (right)

tion of generated questions is a topic for our future study.

User Study Findings: Within *Cisco*, questions of types such as '*Causal Consequence/Interpretation*' are reflective of scenarios faced by tech-support engineers who need intricate knowledge during troubleshooting. In contrast, recall and fact-oriented questions pertaining to the types '*Definition/Feature Specification/Comparison/Example*' correspond to knowledge used by marketing and sales executives who need to keep abreast with information on the rapidly developing novel product offerings from *Cisco* as well as competition.

Relying on experts to design training materials for every newly innovated product at varying complexity levels would be time-consuming and costly. This scenario provides the perfect venue for applying LLM-generated questions. We incorporated a quiz-style interface on top of `Pro-QuEST` and showcased it along with a leader board at the recent Cisco Live event[1] held in Melbourne for engaging our potential users. Two sets of quizzes based on content on recent *Cisco* products and general content from Wikipedia articles (listed in Table 4) were used for this demonstration. Screenshots from our quiz interface along with the leaderboard are shown in Figure 7. During the event, we presented random samples of ten LLM-generated questions as our quizzes and scored the participants based on their choice of correct answer and speed. The time limit for each question was set to 45 seconds per question and prizes were offered to participants who topped the leader boards.[9] Overall, we had six and ten participants for the "technical/non-technical" topic quizzes, respectively. The average score was ∼270 for the former and ∼355 for the latter, in line with the expectation that technical questions are more challenging to answer than the non-technical ones. Our quiz generator tool based on `Pro-QuEST` was well-received during the event and has opened up connections for real deployment within *Cisco*.

Dataset	easy%	med.%	hard%
CCNA/Gemini	39.19	2.7	1.35
CCNA/GPT	32.12	5.45	2.42
CCNP/Gemini	42.60	1.18	4.14
CCNP/GPT	34.59	6.92	11.95
CCNA/Book	10.16	9.63	6.42
CCNP/Book	30.61	4.08	4.08
CCNA/Exam	10.5	12.75	9.25
CCNP/Exam	15.91	5.25	18.88

Table 2: Question percentages for difficulty levels

4 Related Work

LLM-based approaches are now state-of-the-art for various internal tasks within companies involving information processing and language generation including automated text correction, summarization, question answering, entity recognition, product reviews evaluation, and customer support chatbots (Palen-Michel et al., 2024; Wulf and Meierhofer, 2024; Zheng et al., 2023; Roumeliotis et al., 2024; Su et al., 2025; Oh, 2024; Song et al., 2021). As LLMs are being rapidly adopted and still evolving, benchmarking has become an active topic of recent research. Representative benchmarks can characterize LLM model capabilities on domain-specific tasks such as reasoning, conversations, programming (Lu et al., 2021; Lin et al., 2022; Srivastava et al., 2022; Chiang et al., 2024), languages (Dalvi et al., 2024; Baucells et al., 2025) as well as aspects such as factuality and hallucination (Bao

[9]Assuming on average a person would need at least 15 seconds to read and answer, the best possible score is ∼667.

et al., 2025; Wang et al., 2025b). Question answering (QA) datasets are widely used for LLM benchmarking studies (Zhong et al., 2020; Fei et al., 2024; Ouyang et al., 2024; Xie et al., 2024; Chen et al., 2025; Guha et al., 2023). Several research works have addressed the creation of specialized QA datasets using LLMs for domains such as law, medicine, and finance (Choi et al., 2025; Xiong et al., 2024; Scaria et al., 2024; Artsi et al., 2024; Camarata et al., 2025) but, to our knowledge, we are the first to investigate MCQG with LLMs in a highly-technical industry context (such as *Cisco*).

5 Conclusions

We presented `Pro-QuEST`, our system for generating quizzes for technology companies such as *Cisco* who operate with highly specialized domain knowledge. Our experiments with `Pro-QuEST`-questions illustrated their practical usefulness for LLM benchmarking as well as provided insights on future QG studies on the topic. We evaluated `Pro-QuEST` by trialing it with in-house sales personnel at a recent marketing event at *Cisco*. `Pro-QuEST` was enthusiastically received for its potential to efficiently create training quizzes for keeping up with the rapidly-evolving product landscape in networking and security markets.

In future, we would like to study the transferability of `Pro-QuEST` to other technical product domains such as embedded systems, and sensor technologies. Though it was not observed in the samples manually examined in this study, there is a possibility of generated questions and keyphrases to be incorrect, involve hallucination, and be overall unusable. Qualitative evaluation of LLM-generated benchmarks and design of more accurate classification models and taxonomies for question-type characterization comprise some of our future research directions.

Acknowledgments

We thank Sarah Yee from Cisco Systems for her initiative and assistance with setting up the user study at the Cisco Live, Melbourne event in 2025.

This research is supported by A*STAR, CISCO Systems (USA) Pte. Ltd and National University of Singapore under its Cisco-NUS Accelerated Digital Economy Corporate Laboratory (Award I21001E0002).

References

Y. Artsi, V. Sorin, E. Konen, B. S. Glicksberg, G. Nadkarni, and E. Klang. 2024. Large language models for generating medical examinations: systematic review. *BMC medical education*.

Forrest Sheng Bao, Miaoran Li, Renyi Qu, Ge Luo, Erana Wan, Yujia Tang, Weisi Fan, Manveer Singh Tamber, Suleman Kazi, Vivek Sourabh, Mike Qi, Ruixuan Tu, Chenyu Xu, Matthew Gonzales, Ofer Mendelevitch, and Amin Ahmad. 2025. FaithBench: A diverse hallucination benchmark for summarization by Modern LLMs. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, pages 448–461, Albuquerque, New Mexico. Association for Computational Linguistics.

Irene Baucells, Javier Aula-Blasco, Iria de Dios-Flores, Silvia Paniagua Suárez, Naiara Perez, Anna Salles, Susana Sotelo Docio, Júlia Falcão, Jose Javier Saiz, Robiert Sepulveda Torres, Jeremy Barnes, Pablo Gamallo, Aitor Gonzalez-Agirre, German Rigau, and Marta Villegas. 2025. IberoBench: A benchmark for LLM evaluation in Iberian languages. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 10491–10519, Abu Dhabi, UAE. Association for Computational Linguistics.

Florian Boudin and Akiko Aizawa. 2025. An analysis of datasets, metrics and models in keyphrase generation. In *Proceedings of the Fourth Workshop on Generation, Evaluation and Metrics (GEM²)*, pages 973–973, Vienna, Austria and virtual meeting. Association for Computational Linguistics.

Troy Camarata, Lise McCoy, Robert Rosenberg, Kelsey R. Temprine Grellinger, Kylie Brettschnieder, and Jonathan Berman. 2025. Llm-generated multiple choice practice quizzes for preclinical medical students. *Advances in Physiology Education*, 49(3):758–763.

Jiawei Chen, Hongyu Lin, Xianpei Han, and Le Sun. 2024. Benchmarking large language models in retrieval-augmented generation.

Rubing Chen, Jiaxin Wu, Jian Wang, Xulu Zhang, Wenqi Fan, Chenghua Lin, Xiaoyong Wei, and Li Qing. 2025. Benchmarking for domain-specific LLMs: A case study on academia and beyond. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, Suzhou, China. Association for Computational Linguistics.

Wei-Lin Chiang, Lianmin Zheng, Ying Sheng, Anastasios N. Angelopoulos, Tianle Li, Dacheng Li, Banghua Zhu, Hao Zhang, Michael I. Jordan, Joseph E. Gonzalez, and Ion Stoica. 2024. Chatbot arena: an open platform for evaluating llms by human preference. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org.

Chanyeol Choi, Jihoon Kwon, Jaeseon Ha, Hojun Choi, Chaewoon Kim, Yongjae Lee, Jy yong Sohn, and Alejandro Lopez-Lira. 2025. Finder: Financial dataset for question answering and evaluating retrieval-augmented generation. *Preprint*, arXiv:2504.15800.

McKinsey Company. 2023. The economic potential of generative ai: The next productivity frontier. *McKinsey Company*.

Fahim Dalvi, Maram Hasanain, Sabri Boughorbel, Basel Mousi, Samir Abdaljalil, Nizi Nazar, Ahmed Abdelali, Shammur Absar Chowdhury, Hamdy Mubarak, Ahmed Ali, Majd Hawasly, Nadir Durrani, and Firoj Alam. 2024. LLMeBench: A flexible framework for accelerating LLMs benchmarking. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 214–222, St. Julians, Malta. Association for Computational Linguistics.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Jingyuan Ma, Rui Li, Heming Xia, Jingjing Xu, Zhiyong Wu, Baobao Chang, Xu Sun, Lei Li, and Zhifang Sui. 2024. A survey on in-context learning. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 1107–1128, Miami, Florida, USA. Association for Computational Linguistics.

Zhiwei Fei, Xiaoyu Shen, Dawei Zhu, Fengzhe Zhou, Zhuo Han, Alan Huang, Songyang Zhang, Kai Chen, Zhixin Yin, Zongwen Shen, Jidong Ge, and Vincent Ng. 2024. LawBench: Benchmarking legal knowledge of large language models.

Sujatha Das Gollapalli, Mouad Hakam, Mingzhe Du, See-Kiong Ng, and Mohammed Hamzeh. 2025. On assigning product and software codes to customer service requests with large language models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 1092–1103, Suzhou (China). Association for Computational Linguistics.

Neel Guha, Julian Nyarko, and Others. 2023. Legalbench: a collaboratively built benchmark for measuring legal reasoning in large language models. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*.

Kristen Howell, Gwen Christian, Pavel Fomitchov, Gitit Kehat, Julianne Marzulla, Leanne Rolston, Jadin Tredup, Ilana Zimmerman, Ethan Selfridge, and Joseph Bradley. 2023. The economic trade-offs of large language models: A case study. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 5: Industry Track)*, pages 248–267, Toronto, Canada. Association for Computational Linguistics.

Jasmina Jovanovska. 2018. Designing effective multiple-choice questions for assessing learning outcomes. *Infotheca - Journal for Digital Humanities*, 18(1).

Jonathan Kim, Anna Podlasek, Kie Shidara, Feng Liu, Ahmed Alaa, and Danilo Bernardo. 2025. Limitations of large language models in clinical problem-solving arising from inflexible reasoning. *Scientific Reports*, 15(1).

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *NeurIPS*.

Yuanyuan Liang, Jianing Wang, Hanlun Zhu, Lei Wang, Weining Qian, and Yunshi Lan. 2023. Prompting large language models with chain-of-thought for few-shot knowledge base question generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4329–4343, Singapore. Association for Computational Linguistics.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. TruthfulQA: Measuring how models mimic human falsehoods. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3214–3252, Dublin, Ireland. Association for Computational Linguistics.

Shuai Lu, Daya Guo, Shuo Ren, Junjie Huang, Alexey Svyatkovskiy, Ambrosio Blanco, Colin Clement, Dawn Drain, Daxin Jiang, Duyu Tang, Ge Li, Lidong Zhou, Linjun Shou, Long Zhou, Michele Tufano, MING GONG, Ming Zhou, Nan Duan, Neel Sundaresan, and 3 others. 2021. CodeXGLUE: A machine learning benchmark dataset for code understanding and generation. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)*.

Rodney Nielsen, Jason Buckingham, Gary Knoll, Ben Marsh, and Leysia Palen. 2008. A taxonomy of questions for question generation.

Jangmin Oh. 2024. Developing a model for extracting actual product names from order item descriptions using generative language models. *IEEE Access*, 12:122695–122701.

Zetian Ouyang, Yishuai Qiu, Linlin Wang, Gerard De Melo, Ya Zhang, Yanfeng Wang, and Liang He. 2024. CliMedBench: A large-scale Chinese benchmark for evaluating medical large language models in clinical scenarios. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 8428–8438, Miami, Florida, USA. Association for Computational Linguistics.

Chester Palen-Michel, Ruixiang Wang, Yipeng Zhang, David Yu, Canran Xu, and Zhe Wu. 2024. Investigating llm applications in e-commerce. *Preprint*, arXiv:2408.12779.

Mubashar Raza, Zarmina Jahangir, Muhammad Bilal Riaz, Muhammad Jasim Saeed, and Muhammad Awais Sattar. 2025. Industrial applications of large language models. *Scientific Reports*.

Lauro Rizzatti. 2025. A closer look at llm's hyper growth and ai parameter explosion.

Konstantinos I. Roumeliotis, Nikolaos D. Tselikas, and Dimitrios K. Nasiopoulos. 2024. Llms in e-commerce: A comparative analysis of gpt and llama models in product review evaluation. *Natural Language Processing Journal*, 6:100056.

Harsh Saini, Md Tahmid Rahman Laskar, Cheng Chen, Elham Mohammadi, and David Rossouw. 2025. LLM evaluate: An industry-focused evaluation tool for large language models. In *Proceedings of the 31st International Conference on Computational Linguistics: Industry Track*, Abu Dhabi, UAE.

Nicy Scaria, Suma Dharani Chenna, and Deepak Subramani. 2024. *Automated Educational Question Generation at Different Bloom's Skill Levels Using Large Language Models: Strategies and Evaluation*, page 165–179. Springer Nature Switzerland.

Shuangyong Song, Chao Wang, Haiqing Chen, and Huan Chen. 2021. An emotional comfort framework for improving user satisfaction in E-commerce customer service chatbots. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Industry Papers*, pages 130–137, Online. Association for Computational Linguistics.

Zirui Song, Bin Yan, Yuhan Liu, Miao Fang, Mingzhe Li, Rui Yan, and Xiuying Chen. 2025. Injecting domain-specific knowledge into large language models: A comprehensive survey. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 25297–25311, Suzhou, China. Association for Computational Linguistics.

Zayne Rea Sprague, Fangcong Yin, Juan Diego Rodriguez, Dongwei Jiang, Manya Wadhwa, Prasann Singhal, Xinyu Zhao, Xi Ye, Kyle Mahowald, and Greg Durrett. 2025. To cot or not to cot? chain-of-thought helps mainly on math and symbolic reasoning. In *The Thirteenth International Conference on Learning Representations*.

Aarohi Srivastava and 1 others. 2022. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *arXiv preprint arXiv:2206.04615*.

Hanchen Su, Wei Luo, Yashar Mehdad, Wei Han, Elaine Liu, Wayne Zhang, Mia Zhao, and Joy Zhang. 2025. LLM-friendly knowledge representation for customer support. In *Proceedings of the 31st International Conference on Computational Linguistics: Industry Track*, pages 496–504, Abu Dhabi, UAE. Association for Computational Linguistics.

Shichao Sun, Ruifeng Yuan, Ziqiang Cao, Wenjie Li, and Pengfei Liu. 2024. Prompt chaining or stepwise prompt? refinement in text summarization. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 7551–7558, Bangkok, Thailand. Association for Computational Linguistics.

Hamed Tanhaei, Payam Boozary, Sogand Sheykhan, Maryam Rabiee, Farzam Rahmani, and Iman Hosseini. 2024. Predictive analytics in customer behavior: Anticipating trends and preferences. *Results in Control and Optimization*, 17:100462.

Bingning Wang, Xiaochuan Wang, Ting Tao, Qi Zhang, and Jingfang Xu. 2020. Neural question generation with answer pivot. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34:9138–9145.

Suqing Wang, Zuchao Li, Shi Luohe, Bo Du, Hai Zhao, Yun Li, and Qianren Wang. 2025a. From parameters to performance: A data-driven study on LLM structure and development. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, Suzhou, China. Association for Computational Linguistics.

Yuxia Wang, Minghan Wang, Hasan Iqbal, Georgi N. Georgiev, Jiahui Geng, Iryna Gurevych, and Preslav Nakov. 2025b. OpenFactCheck: Building, benchmarking customized fact-checking systems and evaluating the factuality of claims and LLMs. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 11399–11421, Abu Dhabi, UAE. Association for Computational Linguistics.

Angelica Willis, Glenn Davis, Sherry Ruan, Lakshmi Manoharan, James Landay, and Emma Brunskill. 2019. Key phrase extraction for generating educational question-answer pairs. In *Proceedings of the Sixth (2019) ACM Conference on Learning @ Scale*, New York, NY, USA. Association for Computing Machinery.

Tongshuang Wu, Michael Terry, and Carrie Jun Cai. 2022. Ai chains: Transparent and controllable human-ai interaction by chaining large language model prompts. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, CHI '22, New York, NY, USA. Association for Computing Machinery.

Jochen Wulf and Jürg Meierhofer. 2024. Utilizing large language models for automating technical customer support. *Preprint*, arXiv:2406.01407.

Chaojun Xiao, Jie Cai, Weilin Zhao, Biyuan Lin, Guoyang Zeng, Jie Zhou, Zhi Zheng, Xu Han, Zhiyuan Liu, and Maosong Sun. 2025. Densing law of llms. *Nature Machine Intelligence*.

Qianqian Xie and 1 others. 2024. Finben: An holistic financial benchmark for large language models. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Guangzhi Xiong, Qiao Jin, Zhiyong Lu, and Aidong Zhang. 2024. Benchmarking retrieval-augmented generation for medicine. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 6233–6251, Bangkok, Thailand. Association for Computational Linguistics.

Zhijian Xu, Yilun Zhao, Manasi Patwardhan, Lovekesh Vig, and Arman Cohan. 2025. Can LLMs identify critical limitations within scientific research? a systematic evaluation on AI research papers. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 20652–20706, Vienna, Austria. Association for Computational Linguistics.

Changlin Yang, Siye Liu, Sen Hu, Wangshu Zhang, Teng Xu, and Jing Zheng. 2023. Improving knowledge production efficiency with question answering on conversation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 5: Industry Track)*, pages 225–234, Toronto, Canada. Association for Computational Linguistics.

Wenli Yang, Lilian Some, Michael Bain, and Byeong Kang. 2025. A comprehensive survey on integrating large language models with knowledge-based methods. *Knowledge-Based Systems*, 318:113503.

Zhiling Zhang and Kenny Zhu. 2021. Diverse and specific clarification question generation with keywords. In *Proceedings of the Web Conference 2021*, WWW '21, page 3501–3511, New York, NY, USA. Association for Computing Machinery.

Jianhua Zhao and Yinjian Jiang. 2010. Categories of questions in an online discussion forum: An analysis. In *2010 5th International Conference on Computer Science Education*, pages 428–431.

Xin Zheng, Tianyu Liu, Haoran Meng, Xu Wang, Yufan Jiang, Mengliang Rao, Binghuai Lin, Yunbo Cao, and Zhifang Sui. 2023. DialogQAE: N-to-n question answer pair extraction from customer service chatlog. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 6540–6558, Singapore. Association for Computational Linguistics.

Haoxi Zhong, Chaojun Xiao, Cunchao Tu, Tianyang Zhang, Zhiyuan Liu, and Maosong Sun. 2020. Jec-qa: A legal-domain question answering dataset. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

A Appendix

A.1 Datasets

The chapters considered from the CCNA certification guide were "*Introduction to TCP-IP Networking , Fundamentals of Ethernet LANs, Implementing Ethernet Virtual LANs, and Spanning Tree Protocol Concepts*" whereas from the CCNP guide, we considered the chapters on "*Cybersecurity Fundamentals, Cryptography Cisco Secure Firewall, and Virtual Private Networks (VPNs)*". For these chapters, the textbook questions correspond to the ones listed under "Do I know this already?" sections of the chapters. For the exam questions from the question bank, we do not have the specific mapping to the topics/chapters. and hence all questions were included in experiments.

The certification guide books also included representative key terms and topics for a given chapter. We evaluated the keyphrases extracted in Step-2 of our pipeline (Figure 1) with LLM prompts (Table 11) against these expert-drafted lists. On average overlap between the keyphrases extracted with GPT-4o and the expert lists overlap about 34% of the time (Table 3). The list of keyphrases from the textbook and GPT-4o are shown for a sample chapter in Table 7.

Mean	Max	STD
0.342	0.857	0.206

Table 3: Overlap between GPT-extracted and expert-crafted keyphrases for chapters in the CCNA certification guide

The documents used for quizzes in the **User Study** are listed in Table 4. Random samples of 10 were administered for each quiz from the overall sets of 30 generated questions for technical/general documents listed above.

A.2 Models and Parameter Settings

Table 8 presents the list of LLMs used in our benchmarking experiments. Our models include both proprietary and open-source options, with parameter sizes ranging from 3B to 120B. For experiments using open-source LLMs (from Meta and Qwen), we used vLLM for inference on 4×H100 80GB GPUs. For question generation, we extracted 5 keyphrases and 3 keytopics per section (Table 11) and restricted the number of questions to 10 in "free" generation –where LLMs are not constrained by the topic/keyphrase during generation (See **NoKP-MCQP** in Table 10).

Technical Topics
1. https://www.cisco.com/c/dam/en/us/products/collateral/routers/secure-routers/8300-series-secure-routers-ds.pdf
2. https://blogs.cisco.com/security/cisco-hybrid-mesh-firewall-better-enforcement-points-smarter-segmentation-multi-vendor-policy
3. https://www.cisco.com/c/dam/en_us/solutions/artificial-intelligence/ai-infrastructure.pdf
General Topics
1. https://en.wikipedia.org/wiki/Australia
2. https://en.wikipedia.org/wiki/Melbourne
3. https://en.wikipedia.org/wiki/Cisco

Table 4: Documents used for our User Study.

A.3 Question Type analysis

The list of question types (Zhao and Jiang, 2010; Nielsen et al., 2008) with examples from *Cisco* datasets are shown in Table 9. For a cost-effective and efficient method to obtain question type information over all our data, we trained a local model as follows. First, labels of question types for the CCNA-Book and CCNA-Exam questions were obtained by prompting GPT-4o LLM in a zero-shot setting using an MCQ formulation ("Which of the following question type from the list best matches..").

We manually checked samples of GPT-assigned labels and found them to be highly accurate. "Silver" data obtained through GPT was used to train a local prompt-tuned model using Flan-T5-large[10]. Model training and inference was performed on a single GPU of an Nvidia Tesla cluster (Linux) machine with 32GB RAM. For assessing the reliability of predicted question types, we manually analyzed random samples of predictions obtained with our trained model for ten questions for each type from CCNP-Exam (Table 1). On this sample, the labels were 73% accurate with a macro-averaged F1 score of 68. Most errors corresponded to 'Feature Specification' questions incorrectly predicted as 'Definition/Causal Consequence/Interpretation' and 'Example' questions incorrectly predicted as 'Verification'. Due to the direction of these errors and the small percentages of types such as 'Causal Consequence/Causal/Antecedent/Interpretation' in our datasets, we posit that our analysis based on the relative trends of dominant question-type frequencies is still legitimate.

[10] https://huggingface.co/google/flan-t5-large

System Prompt: You are an efficient PDF parser. From the initial content of the PDF provided, extract the metadata requested for. **Sections Prompt**: Does the initial content from a PDF include a listing of topics in in the document? If yes, return a Python list of strings, each string being topic title. Content: []

Table 5: Prompts of extracting metadata/content listing

System Prompt: You are a Cisco technical support engineer with in-depth knowledge of Cisco certification materials. Answer the following multiple-choice question. Only print your answers as a Python list . . . **User Prompt**: Question with Options

Table 6: MCQA Prompts

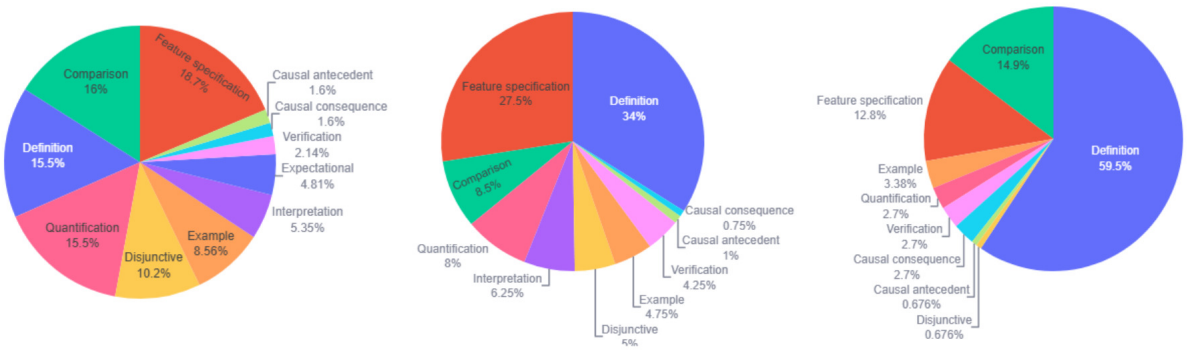

Figure 6: Question Type Distribution for CCNA datasets

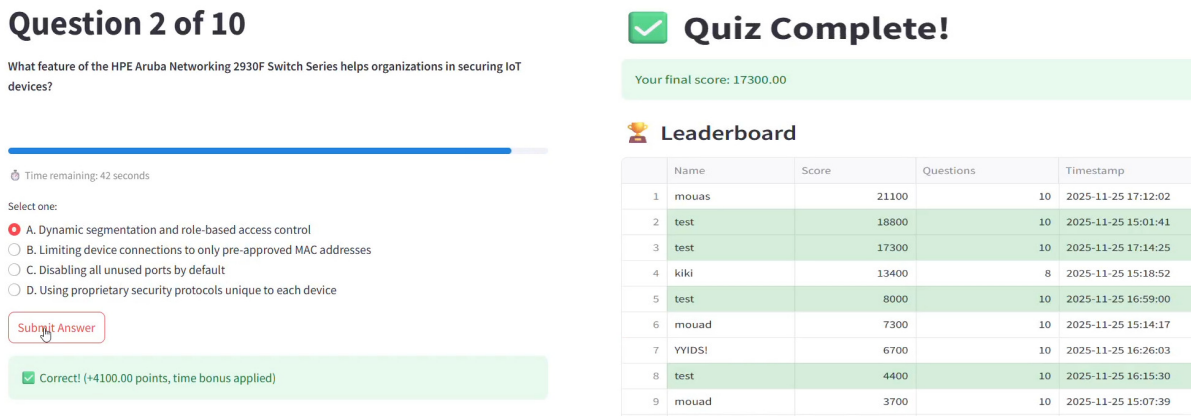

Figure 7: The Pro-QuEST interface shown in Figure 1 is followed by an option to try out the generated questions as a quiz and quiz takers are ranked on a leader board

Gold KPs: 'alternate port (role)', 'backup port (role)', 'blocking state', 'BPDU Filter', 'BPDU Guard', 'bridge ID', 'bridge protocol data unit (BPDU)', 'broken state', 'designated port', 'designated port (role)', 'disabled port (role)', 'disabled state', 'discarding state', 'EtherChannel', 'forward delay', 'forwarding state', 'Hello BPDU', 'learning state', 'listening state', 'Loop Guard', 'MaxAge', 'PortFast', 'Rapid STP (RSTP)', 'root cost', 'Root Guard', 'root port (role)', 'root switch', 'Spanning Tree Protocol (STP)', 'superior BPDU', 'unidirectional link'
GPT KPs: '**Spanning Tree Protocol (STP)**', 'Rapid Spanning Tree Protocol (RSTP)', 'IEEE 802.1D', 'IEEE 802.1w', '**BPDU (Bridge Protocol Data Units)**', 'root switch election', 'root port (RP)', 'designated port (DP)', 'alternate port (ALT)', '**backup port**', 'port roles', 'port states', 'convergence', '**forwarding state**', '**blocking state**', '**listening state**', '**learning state**', '**BPDU Guard**', '**Root Guard**', '**Loop Guard**'

Table 7: Key terms from the textbook and GPT-4o extracted key terms are shown for the chapter 9 from CCNA guide/Chapter 9. Spanning Tree Protocol Concepts

Model Name	Type	Size	Link
Qwen3-4B	Open Source	4B	https://huggingface.co/Qwen/Qwen3-4B-Instruct-2507
Llama-3.1-8B	Open Source	8B	https://huggingface.co/meta-llama/Meta-Llama-3.1-8B-Instruct
Qwen3-30B-A3B	Open Source	30.5B (MoE)	https://huggingface.co/Qwen/Qwen3-30B-A3B-Instruct-2507
GPT-OSS-120B	Open Source	117B (MoE)	https://huggingface.co/openai/gpt-oss-120b
GPT-4o	Proprietary	Unknown	https://openai.com/api/
Gemini 2.5 Flash	Proprietary	Unknown	https://aistudio.google.com/

Table 8: List of selected LLMs used in our system, including their type (proprietary or open source), parameter size (if available), and source links.

Question Type	Description with Examples
`Verification`	For yes/no responses to factual questions ***Example:*** *Which of the following access-list commands, taken from a router's running-config file, match all packets sent from hosts in subnet* []
`Disjunctive`	Questions that require a simple decision between two alternatives. ***Example:*** *Which of the following things are bound together when a new WLAN is created? (Choose two answers.)*
`Feature specification`	Determines qualitative attributes of an object or situation. ***Example:*** *The Wi-Fi Alliance offers which of the following certifications for wireless devices that correctly implement security standards?*
`Quantification`	Determines quantitative attributes of an object or situation. ***Example:*** *What is the maximum number of distribution switches that can be deployed within a hierarchical LAN design building block?*
`Definition`	Determine meaning of a concept. ***Example:*** *Which one of the following is the data encryption and integrity method used by WPA2?*
`Example`	Request for instance of a particular concept or even type. ***Example:*** *Which one of the following is an example of a AAA server?*
`Comparison`	Identify similarities and differences between two or more objects. ***Example:*** *Which answer best compares named standard IP ACLs with numbered standard IP ACLs?*
`Interpretation`	A description of what can be inferred from a pattern of data ***Example:*** *Upon receipt of a configuration BPDU with the topology change flag set, how do the downstream switches react?*
`Causal antecedent`	Asks for an explanation of what state or event causally led to the current state and why. ***Example:*** *What is the main reason SD-Access uses VXLAN data encapsulation instead of LISP data encapsulation?*
`Causal consequence`	Asks for explanation of consequences of event/state ***Example:*** *What happens to a switch port when a BPDU is received on it when BPDU guard is enabled on that port?*
`Expectational`	Asks about expectations or predictions (including violation of expectation) ***Example:*** *Which action would you expect to be true of a router CLI interaction that is not true . . .*
`Judgmental`	Asks about value placed on an idea, advice, or plan. ***Example:*** *. . . Which one of the following things should you do to determine the root cause of her problem?*

Table 9: Question types (Zhao and Jiang, 2010; Nielsen et al., 2008) with examples from *Cisco* datasets

System Prompt: You are a Cisco technical support engineer with in-depth knowledge of CCNA certification materials.
MCQPrefix: Generate exactly one multiple-choice question based on the content provided with no explanation. Return only a JSON-tuple= . . .

MCQP1: **MCQPrefix**. Use the keyphrase [KP]. Content: []
MCQP2: **MCQPrefix**. The keyphrase [KP] must be an answer to the question. Content: []
MCQP3: **MCQPrefix**. The keyphrase [KP] must be must be one of the options for the multiple-choice question . . . Content: []

NoKP-MCQP: Generate exactly [NUM] multiple-choice questions based on the content . . .

KT-TYPEDMCQP: Generate exactly [NUM] multiple-choice questions of type [QTYPE] based on the content provided . . . for the topic [TOPIC]

Table 10: MCQ Prompts

System Prompt for Key Terms: You are a Cisco technical support engineer with in-depth knowledge of CCNA certification materials and are expert in identifying important concepts related to Cisco domain

System Prompt for Key Topics: . . .expert in identifying key topics in the content involving the principles, algorithms, methods, and techniques related to Cisco domain. For example, some key topics can be described as: "Commands to find access ports and assigned VLANs". . .

Prompt-1: For the given passage extract the top-[X%] relevant conceptual keyphrases. Only return your output as a Python list of strings. Passage: [INPUT-PASSAGE]

Prompt-2: Group the given sets of conceptual keyphrases, and select the top-[X%] most important conceptual keyphrases, given the topic: %s. Only return your output as a Python list of strings. List of keyphrases [OUTPUT-FROM-Prompt-1]

Table 11: Keyphrase Prompts

Prompt Type	**Sample Generated Question**
MCQP1	What does the BPDU Guard feature do when it receives a BPDU on a port configured with PortFast?
MCQP2	Which optional STP feature helps prevent forwarding loops by disabling a port if it receives BPDUs on a port that should only connect to endpoint devices
MCQP3	Which feature disables a port if it receives any BPDUs, helping to prevent forwarding loops when unexpected switches connect to access ports?
KT-MCQP	What is the primary function of BPDU Guard in a network configuration?
KT-MCQP/Quantification	What is the default Hello time interval for BPDU in STP
KT-MCQP/Interpretation	Based on the BPDU Guard logic, what can be inferred when a nonroot switch stops receiving Hello BPDUs on its root port?
KT-MCQP/Causal Antecedent	What could cause a switch to start changing the STP topology?

Table 12: Sample generated questions are shown for the key term: **`BPDU Guard`** and key topic: **`Basic logic for BPDU`**